# CA Single Sign-On - 12.52 SP1

## Installing

Date: 13-Jul-2017

# Table of Contents

# Installing

The content in this section describes how to install and configure all the CA Single Sign-On components. Use the Table of Contents to access the content.

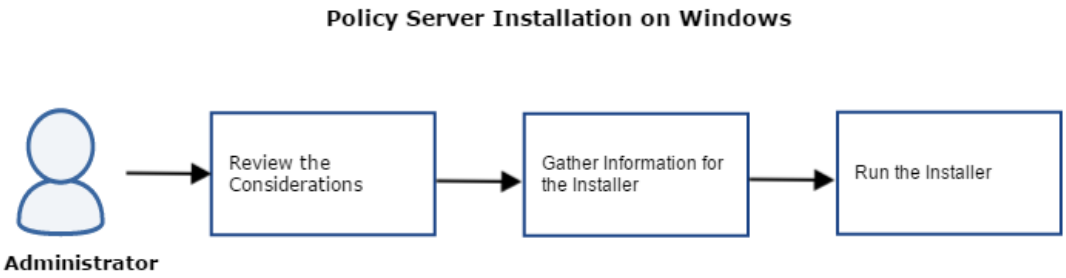
# Install a Policy Server

The following sections describe how to install a Policy Server on Windows and UNIX, how to use the configuration wizard to configure a policy store, and how to configure other supported features.



# Install Policy Server on Windows

The following flowchart describes how to install Policy Server on Windows:




Perform the following steps to install Policy Server:

# Review the Considerations

Review the following considerations before you install Policy Server:

- **Administrator privileges**—A Windows account with local administrator privileges is required to install Policy Server.
- **CPU**—x86 or x64.
- **Memory**—2 GB of system RAM.

  Tip: Use 2 GB of RAM for Policy Server processing, and make available at least 4 GB of RAM to the Policy Server host system.

- **Available disk space**:
  - 4 GB of free disk space in the install location.
  - 3 GB of free space in the temporary file location of the system. These requirements are based on a medium size policy database of approximately 1,000 policies.
- **JRE**—Verify that the required JRE is installed on the Policy Server host system. Verify the supported Java version on the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).
- **JCE**—The current Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction patches are required. To locate the JCE package for your operating platform, go to the Oracle website. Apply the patches to the following files on your system:
  - local_policy.jar
  - US_export_policy.jar

  These files are in the directory <jdk>*jre_home*\lib\security.
  *jre_home*
  This variable specifies the location of the Java Runtime Environment installation.
- **LDAP directory server or relational database**—Verify that you are using a supported LDAP directory server or relational database as a policy store.
- **Windows Firewall**–We recommend that you *disable* Stealth Mode as it increases the time that it takes for agents to make new connections to the Policy Server. These delays can adversely effect functionality such as failover.

- **Firewall settings**—Update the Windows firewall settings to allow inbound connections on the following ports:
  - 44441
  - 44442
  - 44443

  These ports are the default Policy Server accounting, authentication, and authorization ports. If you change these ports after installing Policy Server, allow inbound connections to the respective ports. For more information, see the Microsoft documentation.
- **Environment variables**—The Policy Server installation modifies environment variables.For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix ( http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

# Gather Information for the Installer

The Policy Server installer requires the following information:

- FIPS mode
- Features to be installed and their related configuration information
- JRE location
- Policy Server installation location
- Encryption key value to secure communication between Policy Server and policy store
- Advanced Authentication server encryption key to secure communication between Policy Server and Advanced Authentication server

Complete the following steps to gather the required information to run the installer:

## Determine the FIPS Mode

Policy Server uses certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries. FIPS is a US government computer security standard that is used to accredit cryptographic modules that meet the Advanced Encryption Standard (AES). The libraries provide a FIPS mode of operation when a CA Single Sign-On environment uses only FIPS-compliant algorithms to encrypt sensitive data.

You can install Policy Server in one of the following FIPS modes of operation.

**Note**: The FIPS mode a Policy Server operates in is system-specific. For more information, see the Platform Support Matrix.

- FIPS-compatibility mode—The default FIPS mode of operation during the installation is FIPS-compatibility mode. In FIPS-compatibility mode, the environment uses existing algorithms to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On.
  **Note**:
  - The use of FIPS-compliant algorithms in your environment is optional.
  - If your organization does not require the use of FIPS-compliant algorithms, install Policy Server in FIPS-compatibility mode. No further configuration is required.
- FIPS-migration mode—FIPS-migration mode lets you transition an environment running in FIPS-compatibility mode to FIPS-only mode. In this mode, Policy Server continues to use existing encryption algorithms as you migrate the environment to use only FIPS-compliant algorithms. Use this mode if you are in the process of configuring the existing environment to use only FIPS-compliant algorithms.
- FIPS-only mode—In FIPS-only mode, the environment only uses FIPS-compliant algorithms to encrypt sensitive data. Use this mode if the existing environment is upgraded to a new version and the existing environment is configured to use only FIPS-compliant algorithms.

**Important!** An environment that is running in FIPS-only mode cannot operate with versions of CA Single Sign-On that do not also fully support FIPS (that is, versions before r12.0). This restriction applies to all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. Relink all such software with the versions of the respective SDKs to achieve the required FIPS support.

## Determine the Features to be Installed and Configured

In addition to Policy Server, the installer lets you install and configure the following components:

- OneView Monitor
  Enables monitoring the CA Single Sign-On components. To use the OneView Monitor, you must have the supported Java SDK and Apache Tomcat server installed on the system. Gather the following information:

  - JDK path
    Defines the path to the required JDK version.

  - Apache Tomcat installation directory
    Defines the path to the Apache Tomcat installation directory. If you have multiple Tomcat instances, determine the instance to which you want to configure the OneView Monitor GUI.

  - Tomcat container port number
    Defines the port number for the Tomcat instance. If installing OneView Monitor on the same system as the Administrative UI, change the Tomcat port number from its default 8080 to any other value to avoid a conflict.

  - Sun Java System administrator directory
    Defines the installation location of the Sun Java System and Sun Java System Web servers.

  **Important!** Do not configure OneView Monitor if you are installing Policy Server for the first–time on this system. The installer modifies the configuration files of the web server that is to host the UI. The smuser account does not have the required root privileges. Use the Policy Server Configuration Wizard as the root user to configure OneView Monitor UI after you install Policy Server on the system.

- Web Servers
  Lets you configure to connect your Policy Server with FSS UI to manage single sign-on objects.

- SNMP
  Enables many operational aspects of the environment to be monitored by SNMP-compliant network management applications. You must have the following items to enable SNMP support:

  - The password of the root user

  - A native SunSolstice Master Agent

- Policy store
  The installer can automatically configure one of the following stores as a policy store:

  - Relational Database

  - ADAM/Microsoft Active Directory Lightweight Directory Services (AD LDS)

  - Oracle® Directory Server

## Determine the Policy Store Type

You can configure a policy store using the installer during the Policy Server installation or manually after the Policy Server installation. The installer can automatically configure one of the following stores as a policy store:

- Relational Database
  - Microsoft SQL
  - Oracle
  - PostgreSQL
- ADAM/Microsoft Active Directory Lightweight Directory Services (AD LDS)
- Oracle® Directory Server

Important! For AD LDS and Oracle Directory Server, the installer cannot automatically configure a policy store that is being connected to using an SSL connection.

Review the following considerations:

- (Relational Database) The installer uses specific database information to create the policy store data source with the name CA SiteMinder DSN. Policy Server uses the data source to communicate with the policy store. The installer saves the data source to the system_odbc.ini file that is located in siteminder_home/db.
- (Relational Database) Verify that the database server that is to host the policy store is configured to store objects in UTF–8 form. This configuration avoids possible policy store corruption.
- (Oracle) Oracle supports unicode within many of their character sets. For information about configuring your database to store objects in UTF–8 form, see your vendor–specific documentation.
- (SQL Server) Ensure that the database is configured using the default collation (SQL_Latin1_General_CP1_CI_AS). Using a collation that is case–sensitive can result in unexpected behaviors. For information about configuring your database to store objects using the default collation, see your vendor–specific documentation.
- The key store and certificate data store are automatically configured and collocated with the policy.

The installer prompts for information depending on the database that you select. For information that is required for each database, see the *Gather Database Information* section of the following topics:

- Microsoft SQL Server (see page 242)

- Oracle (see page 254)
- PostGreSQL (see page 230)
- Active Directory (see page 55)
- Oracle Directory Server (see page 112)

## Gather the JRE Location

Gather the location where the supported JRE that is shipped with the JDK is installed.

## Determine the Install Location

Determine where the installer must install Policy Server.

Important! We recommend that you do not exceed 700 characters. The installation fails if the system path length exceeds 1024 characters. The limitation applies to both included or excluded CA Single Sign-On added directories.

## Determine the Encryption Key Value

Determine the encryption key value that secures the data communicated between Policy Server and the policy store. All Policy Servers that share a policy store are required to use the same encryption key. For stronger protection, define a long encryption key. The encryption key is case-sensitive and can contain alphanumeric key value.

**Value**: 6 to 24 characters

## Advanced Authentication Server Encryption Key Value

Determine the encryption key value that secures communication between Policy Server and Advanced Authentication Server, which runs on CA Access Gateway. Use a case–sensitive, alphanumeric value. CA Access Gateway and all Policy Servers require the same key. For stronger protection, define a long encryption key.
**Value**: 6 to 24 characters.

# Run the Installer

Install Policy Server using the installation media. For a list of installation media names, see the *Policy Server Release Notes*.

**Follow these steps:**

1. Exit all applications that are running.
2. Ensure that you have the local administrator privileges to run the installer

3. Perform one of the following steps:
   **Windows 2008**: Right-click *installation_media* and select **Run as administrator**.
   **Other Windows versions**: Double–click *installation_media*.
   *installation_media s*pecifies the name of the Policy Server installation executable.
   The installer starts.

4. Accept the license agreement.

5. Use the information that you gathered in Gather Information for the Installer section to continue working with the installer. Consider the following items when you specify the inputs:

   - You can configure a policy store now or after the Policy Server installation.
   - If you configured a policy store now and are initializing a policy store, you are prompted to enter a password for the default user account. The default account name is siteminder. Initialize the policy store only when you configure a new policy store instance.
   - You are prompted to install the default certificate authority certificates to the certificate data store. You can add additional certificates and private keys to the certificate data store after the installation.
   - If you are using IPv6 addresses, surround entries with brackets.
     **Example**: [2001:db8::1428:57ab]
   - Create an encryption key for the Advanced Authentication server. Use the same key on all Policy Servers in your environment.

6. Review the installation summary and Click **Install**.
   The installation can take several minutes. Policy Server and the selected features, if any, are installed and configured.

7. Exit the installer.

If you experience problems during the installation, you can locate the installation log file and the policy store details file at *siteminder_home*/siteminder/install_config_info.

If you did not use the installer to configure a policy store, run the Policy Server Configuration Wizard (see page 48) to manually configure a policy store. If you configured a policy store, proceed to install Administrative User Interface (see page 386).

## Enable SNMP Event Trapping

**(Only for SNMP)**

**Follow these steps**:

1. Ensure thhat you have an SNMP Service installed on the Windows systems.

2. Use the XPSConfig utility to set the event handler library, eventsnmp.dll, to the XPSAudit list.
   **Default Location of eventsnmp.dll**: *policy_server_home*\bin.

3. Configure the snmptrap.conf file. For information about the necessary SNMP prerequisites and procedures, see SNMP Support (see page 1007).

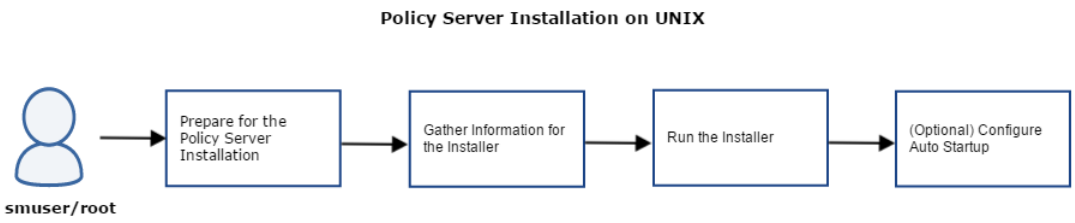
# (Optional) Reinstall the Policy Server

You can reinstall Policy Server over an existing Policy Server of the same version to restore lost application files or restore the Policy Server default installation settings.
**Follow these steps**:

1. Stop Policy Server using the Policy Server Management Console.
2. Close the Policy Server Management Console.
3. Install Policy Server.
4. Start the Policy Server using the Policy Server Management Console.

# Install Policy Server on UNIX

The following flowchart describes how to install Policy Server on UNIX:




**Follow these steps**:

1. Prepare your system for the Policy Server installation (see page 35).
2. Gather information for the installer (see page 40).
3. Run the installer (see page 44).
4. (Optional) Configure auto startup (see page 47).

# Prepare for the Policy Server Installation

Complete the following steps to prepare your system for the Policy Server installation:

- Review Installation Considerations (see page 36)
- Review Global Zone Support on Solaris (see page 37)
- Install Required KornShell Package (see page 38)
- Create a UNIX Account (see page 38)
- Increase Entropy (see page 38)
- Verify Required Linux Libraries (see page 39)

## Review Installation Considerations

Review the following considerations before you prepare your system for the installation:

- **CPU**
  **Solaris**–UltraSparc 440MHz or higher
  **Red Hat**–x86 or x64

- **Memory**—2 GB of system RAM.
  **Tip:** Use 2 GB of RAM for Policy Server processing, and make at least 4 GB of RAM available to the Policy Server host system.

- **Available disk space**

  - 4 GB of free disk space.

  - 3 GB of free disk space in /tmp for Policy Server and 150 MB for Policy Server Configuration Wizard

  Typically, 10 MB of free disk space in /tmp is required for the daily operation of the Policy Server. The Policy Server creates files and named pipes under /tmp. You cannot change the path.

- **JRE**—Verify that the required JRE, which is shipped with JDK, is installed on the Policy Server host system.

- **JCE**—The current Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction patches are required to use the Java cryptographic algorithms. To locate the JCE package for your operating system, go to the Oracle website (http://www.oracle.com/index.html) and download the JCE package for the Java version supported by the Policy Server. Navigate to the directory *jre_home* \lib\security on your system and apply the patches to the following files:

  - local_policy.jar

  - US_export_policy.jar

- **LDAP directory server or relational database**—Verify that you are using a-supported LDAP directory server or relational database as a policy store. For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

- **Exceed X-windows application**—The use of Exceed X-windows application to run the installer or configuration wizard can truncate text in the window because of unavailable fonts in Exceed. This limitation does not impact the installation or configuration.

- **Environment variables**—The Policy Server installation modifies environment variables.

# Review Global Zone Support on Solaris

**(Solaris)**

Consider the following scenarios when planning to run one or more Policy Servers in a Solaris 10 environment. A global zone configuration limits the implementation to a single Policy Server instance across all zones. Specifically:

- Only a single Policy Server instance is supported on the global zone.
- A Policy Server instance is not supported on a sparse-root zone if there is another Policy Server instance on the global zone.
- A Policy Server instance is not supported on a whole-root zone if there is another Policy Server instance on the global zone.

**Note:** Web Agents may run concurrently in any zone.

## Sparse-root Zone Support

A sparse-root zone configuration supports multiple Policy Server instances running on multiple sparse-root zones. Specifically:

- Only one Policy Server instance is supported on each sparse-root zone.
- Concurrent Policy Server instances are supported on sparse-root zones and whole-root zones, so long as there is only one Policy Server instance on each sparse-root or whole-root zone.
- Policy Server instances are not supported running concurrently on the global zone and on sparse-root zones.

**Note:** Web Agents may run concurrently in any zone.

## Whole-root Zone Support

A whole-root zone configuration supports multiple Policy Server instances running on multiple whole-root zones. Specifically:

- Only one Policy Server instance is supported on each whole-root zone.
- Concurrent Policy Server instances are supported on whole-root zones and sparse-root zones, so long as there is only one Policy Server instance on each whole-root zone or sparse-root zone.
- Policy Server instances are not supported running concurrently on the global zone and on whole-root zones.

**Note:** Web Agents may run concurrently in any zone.

## Install Required KornShell Package

Verify that the required library is available on your system:

- Red Hat 5.x 32-bit—ksh-20100621-12.el5.i386.rpm
- Red Hat 5.x 64-bit—ksh-20100621-12.el5.x86_64.rpm
- Red Hat 6.x 32-bit—ksh-20100621-16.el6.i686.rpm
- Red Hat 6.x 64-bit—ksh-20100621-16.el6.x86_64.rpm

## Create a UNIX Account

We recommend that you create a non-privileged (non-root) user account for installing Policy Server with a specific account. Use the default KornShell to create a user account with the name smuser.

To install Policy Server as the root user and then to run it as smuser, change the ownership of the Policy Server binaries to smuser and launch the binaries from the smuser account.

## Increase Entropy

By default, Red Hat uses entropy that is obtained from general computing operations, which generate random numbers. The random numbers that the Red Hat default random number generator generates are available using the following character devices:

- /dev/random. This is the most secure device as it stops supplying numbers when the entropy amount is insufficient for generating a good random output.
- /dev/urandom. This reuses the entropy pool of the kernel and supplies unlimited pseudo-random numbers with less entropy.

Policy Server uses the /dev/random character device for key generation. However, as /dev/random stops supplying numbers when entropy is insufficient, it might impact the Policy Server run time performance.

To increase the source of randomness for the entropy pool, use one of the following options:

- Most secure and FIPS compliant: Install a hardware entropy generator and configure the rngd daemon to use it to populate /dev/random.
  Example: rngd -r /dev/*device_name* -o /dev/random -b
  ***device_name*** is the character device in use. The device name varies depending on the hardware random number generator that you are using, for example, /dev/hwrng.
  For information about the rngd daemon, see the Red Hat documentation.
- Good security and not FIPS compliant: Configure the rngd daemon to populate /dev/random. Execute the following command:

```
rngd -r /dev/urandom -o /dev/random -b
```

Third-party alternatives to the rngd entropy daemon are also available.

- Least secure and not FIPS compliant: Configure a symbolic link between /dev/urandom and /dev /random. Execute the following commands:

```
mv /dev/random /dev/random.org
ln -s /dev/urandom /dev/random
```

**Important!** To ensure that sufficient entropy is available for Policy Server after a system crash or reboot, add your chosen option to an appropriate startup or service script.

To monitor the entropy on the system, execute the following command:

```
watch -n 1 cat /proc/sys/kernel/random/entropy_avail
```

## Verify Required Linux Libraries

If your Red Hat environment supports X11, no additional library files are required for Policy Server. If your Red Hat environment does not support X11 (for example, a headless environment), install the libraries that are required to support X11.

## Modify the Default Limit Parameters

Policy Server opens multiple sockets and files when it is under load. To ensure that the default limit parameters can handle the load, modify the default limit
parameters to avoid resource issues.

To view the default limit parameters, type the following command in a shell window:

```
ulimit -a
```

A list of parameters is displayed.

Example:

| `$ ulimit -a` | |
|---|---|
| `time(seconds)` | `unlimited` |
| `file(blocks)` | `unlimited` |
| `data(kbytes` | `2097148` |
| `stack(kbytes)` | `8192` |
| `coredump(blocks)` | `unlimited` |
| `nofiles(descriptors)` | `256` |

| | |
|---|---|
| vmemory(kbytes) | unlimited |

In the example, the nofiles parameter defines the number of open connections Policy Server allows for the smuser account. The default value is 256. If the
value is not set high enough, Policy Server returns numerous socket errors. So, place the following command in the profile file of smuser account to change this
value:

```
ulimit -n value
```

Example:

```
ulimit -n 1024
```

### Unset Localization Variables

Use of the LC_* environment variables is not permitted. Unset them in the profile of the user account, smuser or root, before you install Policy Server.

### Unset the LANG Environment Variable or Set it to US English

Complete *one* of the following actions:

- Unset the $LANG environment variable. Add the unset LANG command to the profile of the smuser account.
- Set the $LANG environment variable to **en_US**.

## Gather Information for the Installer

The Policy Server installer requires the following information:

- FIPS mode
- Features to be installed and their related configuration information
- JRE location
- Policy Server installation location
- Encryption key value to secure information between Policy Server and policy store

Complete the following steps to gather the required information to run the installer:

- Determine the FIPS Mode (see page 41)
- Determine the Features to be Installed and Configured (see page 41)
- Determine the Policy Store Type (see page 42)
- Gather the JRE Location (see page 44)
- Determine the Install Location (see page 44)
- Determine the Encryption Key Value (see page 44)

## Determine the FIPS Mode

Policy Server uses certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries. FIPS is a US government computer security standard that is used to accredit cryptographic modules that meet the Advanced Encryption Standard (AES). The libraries provide a FIPS mode of operation when a CA Single Sign-On environment uses only FIPS-compliant algorithms to encrypt sensitive data.

You can install Policy Server in one of the following FIPS modes of operation.

**Note**: The FIPS mode a Policy Server operates in is system-specific. For more information, see the Platform Support Matrix.

- FIPS-compatibility mode—The default FIPS mode of operation during the installation is FIPS-compatibility mode. In FIPS-compatibility mode, the environment uses existing algorithms to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On.
  **Note**:
  - The use of FIPS-compliant algorithms in your environment is optional.
  - If your organization does not require the use of FIPS-compliant algorithms, install Policy Server in FIPS-compatibility mode. No further configuration is required.
- FIPS-migration mode—FIPS-migration mode lets you transition an environment running in FIPS-compatibility mode to FIPS-only mode. In this mode, Policy Server continues to use existing encryption algorithms as you migrate the environment to use only FIPS-compliant algorithms. Use this mode if you are in the process of configuring the existing environment to use only FIPS-compliant algorithms.
- FIPS-only mode—In FIPS-only mode, the environment only uses FIPS-compliant algorithms to encrypt sensitive data. Use this mode if the existing environment is upgraded to a new version and the existing environment is configured to use only FIPS-compliant algorithms.

**Important!** An environment that is running in FIPS-only mode cannot operate with versions of CA Single Sign-On that do not also fully support FIPS (that is, versions before r12.0). This restriction applies to all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. Relink all such software with the versions of the respective SDKs to achieve the required FIPS support.

## Determine the Features to be Installed and Configured

In addition to Policy Server, the installer lets you install and configure the following components:

- OneView Monitor
  Enables monitoring the CA Single Sign-On components. To use the OneView Monitor, you must have the supported Java SDK and Apache Tomcat server installed on the system. Gather the following information:

- JDK path
  Defines the path to the required JDK version.

- Apache Tomcat installation directory
  Defines the path to the Apache Tomcat installation directory. If you have multiple Tomcat instances, determine the instance to which you want to configure the OneView Monitor GUI.

- Tomcat container port number
  Defines the port number for the Tomcat instance. If installing OneView Monitor on the same system as the Administrative UI, change the Tomcat port number from its default 8080 to any other value to avoid a conflict.

- Sun Java System administrator directory
  Defines the installation location of the Sun Java System and Sun Java System Web servers.

**Important!** Do not configure OneView Monitor if you are installing Policy Server for the first–time on this system. The installer modifies the configuration files of the web server that is to host the UI. The smuser account does not have the required root privileges. Use the Policy Server Configuration Wizard as the root user to configure OneView Monitor UI after you install Policy Server on the system.

- Web Servers
  Lets you configure to connect your Policy Server with FSS UI to manage single sign-on objects.

- SNMP
  Enables many operational aspects of the environment to be monitored by SNMP-compliant network management applications. You must have the following items to enable SNMP support:

  - The password of the root user

  - A native SunSolstice Master Agent

- Policy store
  The installer can automatically configure one of the following stores as a policy store:

  - Relational Database

  - ADAM/Microsoft Active Directory Lightweight Directory Services (AD LDS)

  - Oracle® Directory Server

## Determine the Policy Store Type

You can configure a policy store using the installer during the Policy Server installation or manually after the Policy Server installation. The installer can automatically configure one of the following stores as a policy store:

- Relational Database

  - Microsoft SQL

- Oracle
- PostgreSQL

- ADAM/Microsoft Active Directory Lightweight Directory Services (AD LDS)
- Oracle® Directory Server

Important! For AD LDS and Oracle Directory Server, the installer cannot automatically configure a policy store that is being connected to using an SSL connection.

Review the following considerations:

- (Relational Database) The installer uses specific database information to create the policy store data source with the name CA SiteMinder DSN. Policy Server uses the data source to communicate with the policy store. The installer saves the data source to the system_odbc.ini file that is located in siteminder_home/db.
- (Relational Database) Verify that the database server that is to host the policy store is configured to store objects in UTF–8 form. This configuration avoids possible policy store corruption.
- (Oracle) Oracle supports unicode within many of their character sets. For information about configuring your database to store objects in UTF–8 form, see your vendor–specific documentation.
- (SQL Server) Ensure that the database is configured using the default collation (SQL_Latin1_General_CP1_CI_AS). Using a collation that is case–sensitive can result in unexpected behaviors. For information about configuring your database to store objects using the default collation, see your vendor–specific documentation.
- The key store and certificate data store are automatically configured and collocated with the policy.

The installer prompts for information depending on the database that you select. For information that is required for each database, see the *Gather Database Information* section of the following topics:

- Microsoft SQL Server (see page 242)
- Oracle (see page 254)
- PostGreSQL (see page 230)
- Active Directory (see page 55)
- Oracle Directory Server (see page 112)

## Gather the JRE Location

Gather the location where the supported JRE that is shipped with the JDK is installed.

## Determine the Install Location

Determine where the installer must install Policy Server.

Important! We recommend that you do not exceed 700 characters. The installation fails if the system path length exceeds 1024 characters. The limitation applies to both included or excluded CA Single Sign-On added directories.

## Determine the Encryption Key Value

Determine the encryption key value that secures the data communicated between Policy Server and the policy store. All Policy Servers that share a policy store are required to use the same encryption key. For stronger protection, define a long encryption key. The encryption key is case-sensitive and can contain alphanumeric key value.

**Limits**: 6 to 24 characters

# Run the Installer

Install Policy Server using the installation media. You can run the installer in GUI mode or console mode. Install the Policy Server directly on the host system. Do not execute the installer across different subnets.

**Note**: To install Policy Server using Telnet or other terminal emulation software, complete the installation using the console mode. If you install in GUI Mode, a Java exception occurs and the installer exits.

**Follow these steps**:

1. Execute the following command to ensure that you have executable permissions to the installation media:

   ```
   chmod +x installation_media
   ```

   ***installation_media***

   Defines the Policy Server installer executable.

2. Exit all applications that are running.

3. Open a shell and navigate to the installation media.

4. Execute the following command:
   **GUI Mode**:

   `./ca-ps-version-unix_version`

   **Console Mode**:

   `./ca-ps-version-unix_version -i console`

   - ***ps-version***

     Specifies the CA Single Sign-On version and, where applicable, cumulative release number.

   - ***unix_version***

     Specifies the UNIX version.

   **Example**: ca-ps-12.52sp1-rhas64-64-cr.bin
   The installer is launched.

5. Accept the license agreement.

6. Use the information that you gathered in Gather Information for the Installer section to continue working with the installer. Consider the following items when you specify the inputs:

   - When prompted, add the environment script to your profile.
   - Do not configure OneView Monitor if you are installing Policy Server for the first–time on this system.
   - You can configure a policy store now or after the Policy Server installation.
   - If you configured a policy store now and are initializing a policy store, you are prompted to enter a password for the default user account. The default account name is siteminder. Initialize the policy store only when you configure a new policy store instance.
   - You are prompted to install the default certificate authority (CA) certificates to the certificate data store. You can add additional certificates and private keys to the certificate data store after the installation.
   - If you are using IPv6 addresses, surround entries with brackets.
     **Example**: [2001:db8::1428:57ab]
   - Create an encryption key for the Advanced Authentication server. Use the same key on all Policy Servers in your environment.

7. Review the installation summary and proceed to Install.
   The installation can take several minutes. Policy Server and the selected features, if any, are installed and configured.

8. Exit the installer.

If you experience problems during the installation, you can locate the installation log file and the policy store details file at *siteminder_home*/siteminder/install_config_info.

If you did not use the installer to configure a policy store, run the Policy Server Configuration Wizard (see page 48) to manually configure a policy store. If you configured a policy store, proceed to install Administrative User Interface (see page 386).

## Configure Security–Enhanced Linux (SELinux) to Work with CA Single Sign-On

SELinux may have one of the following modes assigned to it according to your organization standards.

- **enforcing**
  Specifies that security policy is enforced
- **permissive**
  Prints warnings instead of enforcing the policy
- **disabled**
  Specifies that no SELinux policy is loaded

Check the current status of SELinux and set the mode to either **permissive** or **disabled** to configure it to work with CA Single Sign-On.

**Follow these steps:**

1. Access the **/etc/selinux/config** file.
2. Run the following command to check the current status:

   ```
   sestatus
   ```

3. If SELinux is set to **enforcing**, change the status to either **permissive** or **disabled**.

   ```
   SELINUX=permissive
   or
   SELINUX=disabled
   ```

   Run the following command to switch the SELinux status to permissive for that specific session.

   **setenforce 0**

4. Run the following command to verify the mode that SELinux is currently set to:

   ```
   getenforce
   ```

## (Optional) Add Exceptions to Security–Enhanced Linux (SELinux)

You can add exceptions to SELinux as an additional step but this is optional if you have configured SELinux to work with CA SSO. If Security–Enhanced Linux is enabled on the Policy Server host system, add CA Single Sign-On–exceptions to the environment. Adding the exceptions prevents Security–Enhanced Linux text relocation denials.

**Follow these steps:**

1. Log in to the Policy Sever host system.
2. Open a shell and run the following command:

   ```
   chcon -t textrel_shlib_t /siteminder_home/lib/*
   ```

   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Run the following command:

   ```
   chcon -t textrel_shlib_t /JDK_home/lib/i386/*
   ```

   - **JDK_home**
     Specifies the required JDK installation path.
4. Run the following command:

   ```
   chcon -t textrel_shlib_t /JDK_home/lib/i386/server/*
   ```

   - **JDK_home**
     Specifies the required JDK installation path.

   CA Single Sign-On–specific exceptions have been added.

## Restart the SNMP Daemon

(**Only for SNMP**)

Restart the SNMP daemon if you configured SNMP during the Policy Server installation.

**Follow these steps**:

1. Execute S76snmpdx stop in /etc/rc3.d.
   The SNMP daemon stops.
2. Execute S76snmpdx start in /etc/rc3.d.
   The SNMP daemon starts.

# Configure Auto Startup

You configure auto startup to ensure that the Policy Server restarts automatically when the UNIX system is rebooted.

**Follow these steps:**

1. Modify the S98M script by replacing every instance of the string “nete_ps_root” with an explicit path to the CA Single Sign-on installation directory.
   **Example**: /export/ca/siteminder

2. Change the directory to the siteminder installation directory.

3. Enter **su** and press ENTER**.**

   **Note**: Do not use the suse command.

   You are prompted for a password.

4. Enter the root password and press ENTER.

5. Enter **$ cp S98sm /etc/rc2.d** and press ENTER.
   s98sm automatically calls the stop-all and start-all executables, which stop and start the Policy Server service when the UNIX system is rebooted.

**Note:** If you are using a local LDAP directory server as a policy store, you must configure the LDAP directory to start automatically before starting the Policy Server automatically.

# Run Policy Server Configuration Wizard

You can use Policy Server Configuration Wizard to configure or reconfigure the following features with the same user account that installed Policy Server:

- OneView Monitor
- Web server
- Policy store
- SNMP support

**Important!** If you configured an Oracle iPlanet web server instance for the OneView Monitor or SNMP, do not use the wizard to configure new instances. Configuring new web server instances can cause the existing web server instance to fail.

**Contents**

# Gather Information for the Installer

## OneView Monitor

To use the OneView Monitor, you must have the supported Java SDK and Apache Tomcat server installed on the system. Gather the following information:

- JDK path
  Defines the path to the required JDK version
- Apache Tomcat installation directory
- Defines the path to the Apache Tomcat installation directory. If you have multiple Tomcat instances, determine the instance to which you want to configure the OneView Monitor GUI.
- Tomcat container port number
- Defines the port number for the Tomcat instance. If installing OneView Monitor on the same system as the Administrative UI, change the Tomcat port number from its default 8080 to any other value to avoid a conflict.
- Sun Java System administrator directory
- Defines the installation location of the Sun Java System and Sun Java System Web servers.

## SNMP

You must have the following items to enable SNMP support:

- The password of the root user
- A native SunSolstice Master Agent

## Policy Store

You can manually configure a policy store or use the wizard to automatically configure one of the following stores as a policy store:

- Relational Database
  - Microsoft SQL
  - Oracle
  - PostgreSQL

- ADAM/Microsoft Active Directory Lightweight Directory Services (AD LDS)
- Oracle® Directory Server

**Important!** For AD LDS and Oracle Directory Server, the installer cannot automatically configure a policy store that is being connected to using an SSL connection.

Review the following considerations:

- (Relational Database) The installer uses specific database information to create the policy store data source with the name CA SiteMinder DSN. Policy Server uses the data source to communicate with the policy store. The installer saves the data source to the system_odbc.ini file that is located in siteminder_home/db.
- (Relational Database) Verify that the database server that is to host the policy store is configured to store objects in UTF–8 form. This configuration avoids possible policy store corruption.
- (Oracle) Oracle supports unicode within many of their character sets. For information about configuring your database to store objects in UTF–8 form, see your vendor–specific documentation.
- (SQL Server) Ensure that the database is configured using the default collation (SQL_Latin1_General_CP1_CI_AS). Using a collation that is case–sensitive can result in unexpected behaviors. For information about configuring your database to store objects using the default collation, see your vendor–specific documentation.
- The key store and certificate data store are automatically configured and collocated with the policy.

The installer prompts for information depending on the database that you select. For information that is required for each database, see the *Gather Database Information* section of the following topics:

- Microsoft SQL Server (see page 242)
- Oracle (see page 254)
- PostGreSQL (see page 230)
- Active Directory (see page 55)
- Oracle Directory Server (see page 112)

# Run the Configuration Wizard on Windows

**Follow these steps**:

1. Exit all applications that are running.

2. Navigate to *siteminder_home*\siteminder\install_config_info and double-click ca-ps-config.exe.
   The Policy Server configuration wizard starts.

   **Important!** If User Account Control (UAC) is enabled run the wizard executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

3. Use the system and component information that you gathered to configure a policy store and individual components.

4. Review the configuration summary and click Install.

   The wizard configures the selected components to work with Policy Server. This can take several minutes.

5. Click Done and restart the system.

Proceed to install Administrative User Interface (see page 386).

# Run the Configuration Wizard on UNIX

**Follow these steps**:

1. Exit all running applications.

2. Execute the following script in a ksh shell from the CA Single Sign-On installation directory:

   ```
   . ./ca_ps_env.ksh
   ```

   **Note**: Ensure that there is a space between the periods (. .).

3. Open a shell and execute the following command:
   **GUI Mode**:

   ```
   ./ca-ps-config.bin gui
   ```

   **Console Mode**:

   ```
   ./ca-ps-config.bin -i console
   ```

   The configuration wizard starts.

4. Use the system and component information that you gathered to configure a policy store and individual components.

5. Initialize the policy store instance only to configure a new policy store instance.

6. Review the configuration summary and proceed.
   The wizard configures the selected components to work with Policy Server. This can take several minutes.

7. Exit the wizard once the configuration is complete.

Proceed to install Administrative User Interface (see page 386).

# Configure LDAP Directory Server Policy, Session, and Key Stores

The content in this section describes how to configure LDAP data stores, which include:

- Policy Store (see page 52)
- Session Store (see page 53)
- Key Store (see page 54)

## Policy Store

The CA Single Sign-On policy store is the repository for all policy-related information. All Policy Servers in a CA Single Sign-On installation must share policy store data, either directly or through replication. The Policy Server is installed with tools that let administrators move policy store data from one storage facility to another.

When you install the Policy Server, you can automatically configure one of the following directory servers as a policy store:

- Microsoft Active Directory Lightweight Directory Services (AD LDS)
- Oracle Directory Server (formerly Sun Java System Directory Server)

If you do not use the Policy Server to configure a policy store automatically, manually configure a policy store after installing the Policy Server. After you install the Policy Server, use the Policy Server Management Console to point the Policy Server to an existing policy store.

For a list of supported CA and third-party components, refer to the Platform Support Matrix on the Technical Support site.

**Important!** To avoid policy store corruption, configure the server where the policy store resides to store objects in UTF-8 form. For more information about storing objects in UTF-8 form, see the documentation for that server.

## Default Policy Store Objects Consideration

When you configure a policy store, the following default policy store object files are available:

- smpolicy.xml
- smpolicy-secure.xml

Both files contain the default objects that the policy store requires. If you use the Policy Server Configuration Wizard to configure the policy store automatically, the wizard only uses smpolicy.xml. If you want to use smpolicy-secure.xml, configure the policy store manually.

Both files provide default security settings. These settings are available in the default Agent Configuration Object (ACO) templates that are available in the Administrative UI. The smpolicy-secure file provides more restrictive default security settings. Choosing smpolicy.xml does not limit you from using the more restrictive default security settings. You can modify the default ACO settings using the Administrative UI.

The following table summarizes the security settings for both files:

| Parameter Name | smpolicy Values | smpolicy–secure Values |
|---|---|---|
| BadCssChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadQueryChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadUrlChars | //, ./, /., /*, *., ~, \, %00-%1f, %7f-%ff, %25 | smpolicy.smdif values plus:<br><, >, ', ;, ), (, &, + |
| EnableCookieProvider | Yes | No |
| IgnoreExt | .class, .gif, .jpg, .jpeg, .png, .fcc, .scc, .sfcc, .ccc, .ntc | All smpolicy values. |
| LimitCookieProvider | No | Yes |
| ValidTargetDomain | This file does not include this parameter. | This parameter does not have a default value. Provide a valid redirection domain.<br>Example: validtargetdomain=".example.com" |

# Session Store

The session store is where the Policy Server stores persistent session data. A persistent session is one in which a cookie is maintained in the session store, in the memory of the web browser, and optionally the hard disk. CA Directory is the only LDAP directory server that the Policy Server supports as a session store.

Before you implement persistent sessions, consider the following information:

- Persistent sessions are configured on a per realm basis.

- Use Persistent sessions only when necessary. Using session services to maintain sessions has an impact on system performance.

If you plan to use persistent sessions in one or more realms, enable the session store using the Policy Server Management Console.

# Key Store

The key store holds web agent keys and session ticket key, which are distributed to Agents at run time.

Web Agents use an agent key to encrypt cookies before passing the cookies to a browser. When a Web Agent receives a CA Single Sign-Oncookie, the agent key enables the Agent to decrypt the contents of the cookie. Keys must be set to the same value for all Web Agents communicating with a Policy Server.

The Policy Server and Agents use encryption keys to encrypt and decrypt sensitive data that is passed between Policy Servers and Agents.

- The Agent uses agent keys to encrypt CA Single Sign-On cookies that are read and shared by all agents in a single sign-on environment. The agent key also decrypts cookies encrypted by the other agents. The Policy Server manages agent keys and distributes the keys to agents periodically.

- Session ticket key is used by the Policy Server to encrypt session tickets. Session tickets contains credentials and other information that is related to a session (including user credentials). Agents embed session tickets in CA Single Sign-On cookies, but do not have access to the session ticket key, which never leaves the Policy Server.

Both types of keys are kept in the Policy Server key store and distributed to Agents at runtime. By default, the key store is part of the Policy Store, but if necessary, you can create a separate key store database.

# Configure an LDAP Directory Server as a Policy Store

This section explains the following Policy Store configurations:


- Configure Active Directory as a Policy Store (see page 55)
- Configure a Domino Directory Server as a Policy Store (see page 64)
- Configure an IBM Tivoli Directory Server as a Policy Store (see page 72)
- Configure Microsoft Active Directory LDS as a Policy Store (see page 80)
- Configure Novell eDirectory as a Policy Store (see page 89)
- Configure OpenLDAP as a Policy Store (see page 99)
- Configure an Oracle Directory Server as a Policy Store (see page 112)
- Configure an Oracle Internet Directory Server as a Policy Store (see page 122)
- Configure Oracle Unified Directory as a Policy Store (see page 132)
- Configure Oracle Virtual Directory as a Policy Store (see page 144)
- Configure a Red Hat Directory Server as a Policy Store (see page 151)

## Configure Active Directory as a Policy Store

**Contents**



Active Directory can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

**Note:** If Policy Server is to communicate with the Active Directory over SSL, ensure that the RootCA Certificate of the Active Directory is available on the Policy Server machine in the cert8.db file.

Gather Directory Server Information
Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.

3. Select the following value from the Database list:

   `Policy Store`

4. Select the following value from the Storage list:

   LDAP

5. Configure the following settings in the LDAP Policy Store group box.

   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `LDAP`
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects.

**Important!** The Active Directory schema is owned by the root domain. All schema changes must be made on the root domain controller that holds the Schema Master using an account with SchemaAdmins permissions. Schema changes cannot be made from child domains (replicas).

**Follow these steps:**

1. Run the following command on the Policy Server host system:

   `smldapsetup ldgen -f<file_name>`

   - **file_name**
     Specifies the name of the LDIF file you are creating.

   An LDIF file with the CA Single Sign-On schema is created.
2. Run the following command:

   `smldapsetup ldmod -f<file_name>`

   - **file_name**
     Specifies the name of the LDIF you created.

   The utility imports the policy store schema.
3. Navigate to *policy_server_home*\xps\db and open the following file:

   `ActiveDirectory.ldif`

4. Manually replace each instance of <RootDN> with the DN that represents the policy store schema location, not the policy store object location.
   **Example:** If the following root DN represents the policy store object:

   ```
   ou=policystore,dc=domain,dc=com
   ```

   Replace each instance of <RootDN> with the following DN:

   ```
   dc=domain,dc=com
   ```

5. Run the following command:

   ```
   smldapsetup ldmod -fpolicy_server_home\xps\db\ActiveDirectory.ldif
   ```

   - **policy_server_home**
     Specifies the Policy Server installation path.

   The policy store schema is extended. You have created the policy store schema.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.

- The password cannot include an ampersand (&) or an asterisk (*).
- If the password contains a space, enclose the passphrase with quotation marks.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

```
XPSImport smpolicy.xml -npass
```

- To import smpolicy–secure.xml, run the following command:

  ```
  XPSImport smpolicy-secure.xml -npass
  ```

  - **npass**
    Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

  Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

**Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
   c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   - **passphrase**
     Specifies the password for the default super user account (siteminder).

**Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Support for Active Directory ObjectCategory Indexing Attribute

Unlike other LDAP-compatible directories, Active Directory does not index policy store objects using the objectClass attribute by default. Instead, the objects are indexed by the objectCategory attribute. To enhance searches, you can either configure objectClass as an indexable attribute (see the Active Directory documentation) or enable objectCategory support in the Policy Server.

## Enable or Disable ObjectCategory Attribute Support

**On Windows:**

**Follow these steps:**

1. Launch the Windows Registry Editor.

2. Locate the key
   HKLM\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\DS\LDAPProvider.

   a. To enable support, set the EnableObjectCategory value to 1.

   b. To disable support, set the EnableObjectCategory value to 0.
      The default value is 0.

**On UNIX:**

**Follow these steps:**

1. In a text editor, open the CA Single Sign-On sm.registry file, which is located in *<siteminder_installation>*/registry.

2. Locate the key
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Netegrity\SiteMinder\CurrentVersion\Ds\

a. To enable support, set the EnableObjectCategory value to 1.

b. To disable support, set the EnableObjectCategory value to 0.
The default value is 0.

# Configure a Domino Directory Server as a Policy Store

**Contents**



Domino Directory Server can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.
- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Create the Policy Store Schema

You create the policy store schema so that the Domino directory server can operate as a policy store.

**Follow these steps:**

1. Stop the domino directory service.
2. Create a backup copy of the following file in your Domino directory server:

   *domino_home*\data\schema.nsf
3. Locate the following file on your Policy Server:

   *policy_server_home*\db\tier2\IBM_Lotous_Domino_DirectoryServer\schema.nsf
4. Copy the file from Step 3 to the following folder of your Domino directory server:

   *domino_home*\data
5. Start the Domino directory service.
6. Open the schema.nsf file on your Domino directory server.
7. Verify that all the xps and CA Single Sign-On objects and attributes exist.
8. Use an LDAP browser to connect to the Domino LDAP directory.
9. Create the following base DN for CA Single Sign-On:

   Netegrity/SiteMinder/PolicySvr4
10. Restart the Domino directory service.
11. Open the service in console mode.
12. Update the indices in the directory with the following command:

    load updall -r
13. Restart the domino directory service.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`
4. Select the following value from the Storage list:

   LDAP
5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.
6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure an IBM Tivoli Directory Server as a Policy Store

**Contents**



IBM Tivoli Directory Server can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Create a Directory Entry and Root Nodes

You use the IBM Tivoli Directory Server Web Administration Tool to create a directory entry and root nodes.

**Note:** If applicable, create or load a server suffix using the IBM Tivoli Directory Server Configuration Tool.

**Follow these steps:**

1. Create a directory entry for the root DN of the policy server data.
   **Example:**

   ```
   ou=Nete
   ```

2. Create the following root nodes under the root DN:
   **Example:**

   ```
   ou=Netegrity,ou=SiteMinder,ou=PolicySvr4,ou=XPS
   ```

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.

3. Select the following value from the Database list:

   ```
   Policy Store
   ```

4. Select the following value from the Storage list:

   ```
   LDAP
   ```

5. Configure the following settings in the LDAP Policy Store group box.

- LDAP IP Address
- Admin Username
- Password
- Confirm Password
- Root DN

**Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `LDAP`
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store objects.

**Follow these steps:**

1. Access the directory server using the IBM directory server configuration tool.
2. Navigate to *policy_server_home*\IBMDirectoryServer.
   - **policy_server_home**
     Specifies the Policy Server installation path.
3. Use the IBM directory server configuration tool to add the V3.siteminder*release* schema file to the Manage Schema Files section of the schema configuration.
   - **release**
     Specifies the CA Single Sign-On release.

4. Navigate to *policy_server_home*\xps\db.

5. Locate the following file:
   IBMDirectoryServer.ldif

6. Use the IBM directory server configuration tool to add the file to the Manage Schema Files section of the schema configuration.

7. Restart the directory server.
   The policy store schema is created.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

**Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

```
XPSImport smpolicy.xml -npass
```

- To import smpolicy–secure.xml, run the following command:

  ```
  XPSImport smpolicy-secure.xml -npass
  ```

  - **npass**
    Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

  Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

**Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Run the following command:

   ```
   XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
   c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   - **passphrase**
     Specifies the password for the default super user account (siteminder).

> **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  > **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  > **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.
- **-vI**
  (Optional) Sets the verbosity level to INFO.
- **-vW**
  (Optional) Sets the verbosity level to WARNING.
- **-vE**
  (Optional) Sets the verbosity level to ERROR.
- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure Microsoft Active Directory LDS as a Policy Store

Microsoft Active Directory LDS can function as a policy store. The Policy Server configuration wizard can set up this directory automatically as a policy store. However, if you did not use the wizard for set up, follow these instructions to set up the policy store up manually.

You can use this single directory instance as a policy store and key store. Using a single directory server simplifies administration tasks, but if your implementation requires, you can configure a separate key store.

This topic includes the following procedures:



### Active Directory LDS Prerequisite

Only an administrative user in the configuration partition can import the policy store schema. This user must have administrative rights over the configuration partition and all application partitions, including the policy store partition.

**Follow these steps:**

1. Create an administrative user in the configuration partition
2. Open the ADSI Edit console.
3. Navigate to the following in the configuration partition:

   ```
   cn=directory service, cn=windows nt,
   cn=services, cn=configuration, cn={guid}
   ```

4. Locate the msDS-Other-Settings attribute.
5. Add the following new value to the msDS-Other-Settings attribute:

   ```
   ADAMAllowADAMSecurityPrincipalsInConfigPartition=1
   ```

6. In the configuration and policy store application partitions:
   a. Navigate to CN=Administrators, CN=Roles.
   b. Open the properties of CN=Administrators.
   c. Edit the member attribute.
   d. Click Add DN and paste the full DN of the user you created in the configuration partition in Step 1.
   e. Go to the properties of the user you created and verify the value for the following object:

      ```
      msDS-UserAccountDisabled
      ```

      Be sure that the value is set false.

   The administrative user has rights over the configuration partition and all application partitions, including the policy store partition.

## Gather Directory Server Information

Configuring Active Directory LDS as a policy store requires specific directory server information. Gather the following information before configuring the policy store.

- **Host information**—Determine the fully qualified name or the IP address of the directory server host system.
- **Port information**—Determine if the directory server is listening on a non–standard port. If you do not provide port information, the CA Single Sign-On utilities you use to configure the policy store default to port 389 (non-SSL) and 636 (SSL).
- **Administrator DN**—Determine the full domain name, including the guid value, of the directory server administrative user.
  **Example**: CN=user1,CN=People,CN=Configuration,CN,{guid}
- **Administrator password**—Determine the password for the directory server administrative user.

- **Root DN of the application partition**—Identify the root DN location of the application partition in the directory server where the policy store schema data must be installed.

- (Optional) **SSL client certificate**—If the directory connection is made over SSL, determine the path of the directory that contains the SSL client certificate database.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Note:** The Policy Server can bind to an AD LDS policy store using a proxy object. A proxy object is created on AD LDS and is associated with an Active Directory account through the Security Identifier of the account. For more information about binding to an AD LDS instance using a proxy object, see the Microsoft documentation. If you configure a Policy Server connection using a proxy object and plan on using password policies, configure AD LDS for SSL.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.

3. Select the following value from the Database list:

   `Policy Store`

4. Select the following value from the Storage list:

   LDAP

5. Configure the following settings in the LDAP Policy Store group box.

   - LDAP IP Address

   - Admin Username
     Specify the full domain name, including the guid value, of the directory server administrator.

   - Password

   - Confirm Password

   - Root DN
     Specifies the existing root DN location of the application partition in the AD LDS server. The existing root DN location is where the policy store schema is imported.

> **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following option:

   `Use Policy Store database`
10. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects.

**Follow these steps:**

1. Run the following command:

   `smldapsetup ldgen -f`*`file_name`*

   - *file_name*
     Specifies the name of the LDIF file you are creating.

   An LDIF file with the CA Single Sign-On schema is created.
2. Run the following command:

   `smldapsetup ldmod -f`*`file_name`*

   - *file_name*
     Specifies the name of the LDIF you created.

   smldapsetup imports the policy store schema.
3. Navigate to *siteminder_home*\xps\db and open the following file:

   `ADLDS.ldif`

   - *siteminder_home*
     Specifies the Policy Server installation path.
4. Replace each instance of {guid} with the actual value of guid in braces and save the file.
   **Example:** {CF151EA3-53A0-44A4-B4AC-DA0EBB1FF200}

5. Run the following command:

   ```
   smldapsetup ldmod -fsiteminder_home\xps\db\ADLDS.ldif
   ```

   The policy store schema is extended. You have created the policy store schema.

## Set the Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

```
XPSImport smpolicy-secure.xml -npass
```

- **npass**
  Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
   c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   - **passphrase**
     Specifies the password for the default super user account (siteminder).

     **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure Novell eDirectory as a Policy Store

**Contents**



Novell eDirectory can function as a policy store. A single directory server instance can function as a:

- Policy store

- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

Be sure that you have the following installed before beginning:

- Novell Windows Login Client

- Novell ConsoleOne for Windows, UNIX, and Netware systems

### Limitations of Policy Store Objects in Novell eDirectory

Consider the following items when working with Policy Store objects in a Novell eDirectory:

- Use a policy store root DN no longer than 15 characters.
  A Novell eDirectory DN cannot exceed 256 characters. Some CA Single Sign-On objects can reach 241 characters. If your root DN is longer than 15 characters, some objects can exceed the 256–byte limit.

- When the policy store resides in Novell eDirectory, policy store objects cannot have names longer than 64 characters. eDirectory does not allow an attribute to be set to a value longer than 64. The limitation affects Certificate Maps particularly because they routinely have long names by design.

- The Policy Server does not support LDAP referrals for policy stores residing in Novell eDirectory.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.

- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Edit the Policy Store Schema Files

Edit the Novell policy store schema file to be sure that it contains your Novell server DN information. You edit the Novell policy store schema file from the Novell Client.

**Follow these steps:**

1. Navigate to *policy_server_home*\bin on the Policy Server host system.

   - **policy_server_home**
     Specifies the Policy Server installation path.

2. Run the following command:

```
ldapsearch -hhost -pport -bbasedn -ssub -Dadmin_DN -wAdminPWd
objectclass=ncpServer dn
```

- **-hhost**
  Specifies the fully qualified host name or the IP Address of the directory server.

- **-pport**
  Specifies the port on which the LDAP directory server is listening.

- **-bbasedn**
  Specifies the base DN for the search.

- **-Dadmin_dn**
  Specifies the DN of the administrator account that can bind to the directory server.

- **-wadmin_pw**
  Specifies the password for the administrator account.

**Example**:

```
ldapsearch -h192.168.1.47 -p389 -bo=nwqa47container -ssub
-Dcn=admin,o=nwqa47container -wpassword objectclass=ncpServer dn
```

The Novell server DN opens.

3. Navigate to *policy_server_home*\novell.
4. Open the Novell policy store schema file.
5. Manually edit the policy store schema file by replacing every <ncpserver> variable with the value that you found in step 2 for your Novell server DN.
   **Example:** If your Novell server DN value is cn=servername,o=servercontainer, replace every instance of <ncpserver> with cn=servername,o=servercontainer.
6. Save and close the policy store schema file.
7. Navigate to *policy_server_home*\xps\db.
8. Open the following Novell policy store schema file:

   ```
   Novell.ldif
   ```

9. Manually edit the policy store schema file by replacing every <ncpserver> variable with the value that you found in step 2 for your Novell server DN.
   **Example:** If your Novell server DN value is cn=servername,o=servercontainer, replace every instance of <ncpserver> with cn=servername,o=servercontainer.

The Novell policy store schema files contain your Novell server DN information.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`
4. Select the following value from the Storage list:

   LDAP
5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.
6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects. You use the smldapsetup tool to add the policy store schema.

**Follow these steps:**

1. Run the following command:

   ```
   smldapsetup ldmod -v
   -fpolicy_server_home\novell\Novell_Add_release.ldif
   ```

   - **-fpolicy_server_home**
     Specifies the Policy Server installation path.
   - **-v**
     Turns on tracing and outputs error, warning, and comment messages.
   - **release**
     Specifies the CA Single Sign-On release.

2. Run the following command:

   ```
   smldapsetup ldmod -v -fpolicy_server_home\xps\db\Novell.ldif
   ```

   The policy store schema is created.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

```
smreg -su password
```

- *password*
  Specifies the password for the default administrator.

The password has the following requirements:

- The password must contain at least six (6) characters and cannot exceed 24 characters.
- The password cannot include an ampersand (&) or an asterisk (*).
- If the password contains a space, enclose the passphrase with quotation marks.

**Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

- **siteminder_home**
  Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Refresh the LDAP Server

You refresh the LDAP server to help ensure that the changes take effect on Novell eDirectory. You use the Novell Client to refresh the LDAP server.

**Follow these steps:**

1. Open ConsoleOne.
2. Double-click LDAP server from the directory tree.
3. Click Refresh LDAP Server Now.
   The LDAP server is refreshed.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure OpenLDAP as a Policy Store

**Contents**



OpenLDAP can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Configure the Slapd Configuration File

An OpenLDAP directory server requires additional configuration before you can use it as a policy store. The following process lists the configuration steps:

1. Specify the CA Single Sign-On schema files.
2. Specify policy store indexing.
3. Enable user authentication.
4. Specify database directives.
5. Support Client-Side Sorting
6. Test the configuration file.
7. Restart the OpenLDAP server.

## Specify the CA Single Sign-on Schema Files

Specifying the schema files in the include section of the slapd configuration file (slapd.conf) configures the slapd process (the LDAP Directory Server daemon) to read the additional configuration information. The included files must follow the correct slapd configuration file format.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*/db/tier2/OpenLDAP and copy the following files to the schema folder in the OpenLDAP installation directory:
   - openldap_attribute.schema
   - openldap_object.schema
   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Navigate to *siteminder_home*/xps/db and copy the following files to the schema folder in the OpenLDAP installation directory:
   - openldap_attribute_XPS.schema

- openldap_object_XPS.schema

4. Type the following entries in the include section of the slapd configuration file:

```
....
.....
include /usr/local/etc/openldap/schema/openldap_attribute.schema
include /usr/local/etc/openldap/schema/openldap_object.schema
include /usr/local/etc/openldap/schema/openldap_attribute_XPS.schema
include /usr/local/etc/openldap/schema/openldap_object_XPS.schema
```

**Note**: This procedure assumes that the OpenLDAP server is located in /usr/local/etc /openldap and that the schema files are located in the schema subdirectory.

The CA Single Sign-On schema files are specified.

## Specify Policy Store Indexing

Specify indexing in the slapd.conf file to use OpenLDAP as a policy store.

**Follow these steps:**

1. Stop the slapd instance.
2. Open the slapd.conf file with a text editor.
3. Locate the following lines:

```
# Indices to maintain
index   objectClass     eq
```

4. Insert a new line in the file, and then add the following lines:

```
index smAdminOID4 pres,eq
index smAuthDirOID4 pres,eq
index smAzDirOID4 pres,eq
index smcertmapOID4 pres,eq
index smIsRadius4 pres,eq
index smIsAffiliate4 pres,eq
index smParentRealmOID4 pres,eq
index smPasswordPolicyOID4 pres,eq
index smAgentGroupOID4 pres,eq
index smKeyManagementOID4 pres,eq
index smAgentOID4 pres,eq
index smAgentKeyOID4 pres,eq
index smRootConfigOID4 pres,eq
index smAGAgents4 pres,eq
index smDomainAdminOIDs4 pres,eq
index smDomainOID4 pres,eq
index smvariableoid5 pres,eq
index smNestedVariableOIDs5 pres,eq
index smvariabletypeoid5 pres,eq
index smActiveExprOID5 pres,eq
index smDomainUDs4 pres,eq
index smVariableOIDs5 pres,eq
index smusractiveexproid5 pres,eq
index smPropertyOID5 pres,eq
index smPropertySectionOID5 pres,eq
index smPropertyCollectionOID5 pres,eq
index smFilterClass4 pres,eq
```

```
index smTaggedStringOID5 pres,eq
index smNoMatch5 pres,eq
index smTrustedHostOID5 pres,eq
index smIs4xTrustedHost5 pres,eq
index smDomainMode5 pres,eq
# index smImsEnvironmentOIDs5 pres,eq
index smSecretRolloverEnabled6 pres,eq
index smSecretGenTime6 pres,eq
index smSecretUsedTime6 pres,eq
index smSharedSecretPolicyOID6 pres,eq
index smFilterPath4 pres,eq
index smPolicyLinkOID4 pres,eq
index smIPAddress4 pres,eq
index smRealmOID4 pres,eq
index smSelfRegOID4 pres,eq
index smAzUserDirOID4 pres,eq
index smResourceType4  pres,eq
index smResponseAttrOID4 pres,eq
index smResponseGroupOID4  pres,eq
index smResponseOID4     pres,eq
index smRGResponses4 pres,eq
index smRGRules4     pres,eq
index smRuleGroupOID4 pres,eq
index smRuleOID4   pres,eq
index smSchemeOID4 pres,eq
index smisTemplate4  pres,eq
index smisUsedbyAdmin4 pres,eq
index smSchemeType4   pres,eq
index smUserDirectoryOID4 pres,eq
index smODBCQueryOID4 pres,eq
index smUserPolicyOID4 pres,eq
index smAgentTypeAttrOID4 pres,eq
index smAgentTypeOID4 pres,eq
index smAgentTyperfcid4 pres,eq
index smAgentTypeType4 pres,eq
index smAgentCommandOID4 pres,eq
index smTimeStamp4     pres,eq
index smServerCommandOID4 pres,eq
index smAuthAzMapOID4 pres,eq
index xpsParameter pres,eq
index xpsValue  pres,eq
index xpsNumber pres,eq
index xpsCategory pres,eq
index xpsGUID pres,eq
index xpsSortKey pres,eq
index xpsIndexedObject pres,eq
```

5. Save the file and close the text editor.

6. Run the following command:

   ```
   slapindex -f slapd.conf
   ```

7. Restart the slapd instance.
   The policy store indexing for OpenLDAP is specified.

## Enable User Authentication

Enabling user authentication ensures that you can protect resources with a supported authentication scheme.

To enable user authentication, add the following to the slapd configuration file:

```
access to attrs=userpassword
by self write
by anonymous auth
by * none
```

## Specify Database Directives

The slapd configuration file requires values for additional database directives.

To specify the directives, edit the following:

- **database**
  Specify any supported backend type.
  **Example**: bdb

- **suffix**
  Specify the database suffix.
  **Example**: dc=example,dc=com

- **rootdn**
  Specify the DN of root.
  **Example**: cn=Manager,dc=example,dc=com

- **rootpw**
  Specify the password to root.

- **directory**
  Specify the path of the database directory.
  **Example**: /usr/local/var/openldap-data

  **Note**: The database directory must exist prior to running slapd and should only be accessible to the slapd process.

## Support Client-Side Sorting

OpenLDAP is the only supported LDAP directory that does not support server-side sorting. Instead, OpenLDAP requires that all sorting be performed on the client side. To accomplish this, all XPS objects are retrieved at start-up using server-side paging.

To support client-side sorting, the OpenLDAP directory administrator must configure the following settings in the slapd.conf file:

- Enable reading of the Root DSE.
  This setting allows the XPS client to read the OpenLDAP directory's type and capabilities.

- Set the maximum number of entries that can be returned from a search operation >= 500.
  This setting accommodates XPS objects which are retrieved in increments of 500 by server-side paging.

- Allow a simple V2 bind.
  This setting allows smconsole to test the LDAP connection using a simple V2 bind.

**Follow these steps:**

1. Add the following lines to the slapd.conf file:

```
access to * by users read by anonymous read
```

Or

```
access to dn.base="ou=<Root_DN>" by users read
```

- ***Root_DN***
  Specifies the distinguished name of the node in the LDAP tree where policy store objects must be defined.

**Note:** For more information on how to specify the ACL, see http://www.openldap.org/doc/admin24/access-control.html.

2. Verify that the value specified by the sizelimit directive in the slapd.conf file >= 500:

   ```
   sizelimit 500
   ```

   **Note:** The default sizelimit value is 500. For more information, see http://www.openldap.org/doc/admin24/slapdconfig.html.

3. Add the following line to the slapd.conf file:

   ```
   allow bind_v2
   ```

The slapd.conf file is configured to support client-side sorting.

## Test the Configuration File

Testing the configuration file ensures that it is correctly formatted.

**Follow these steps:**

1. Change the directory to the OpenLDAP server directory.
2. Run the following command:

   ```
   ./slapd
   ```

   **Note**: Unless you specified a debugging level, including level 0, slapd automatically forks, detaches itself from its controlling terminal, and runs in the background.

3. Run the following command:

   ```
   ./slapd -Tt
   ```

   The slapd configuration file is tested.

## Restart the OpenLDAP Server

Restarting the OpenLDAP directory server loads the CA Single Sign-on schema. The Policy Server requires that the CA Single Sign-on schema is loaded before you can use the directory server as a policy store.

**Follow these steps:**

1. Stop the directory server using the following command:

   ```
   kill -INT 'cat path_of_var/run_directory/slapd.pid`
   ```

   - **path_of_var/run_directory**
     Specifies the path of the database directory.
     Example: kill -INT `cat /usr/local/var/run/slapd.pid`

2. Start the directory server using the following command:

   ```
   ./slapd
   ```

## Create the Database

The following process lists the steps for creating the directory server database for the policy store:

1. Create the base tree structure.

2. Add entries.

## Create the Base Tree Structure

You can create a base tree structure to store policy store objects.

Specify the following entry under the root DN:

```
ou=Netegrity,ou=SiteMinder,ou=PolicySvr4,ou=XPS
```

The base tree structure is created.

## Add Entries

Add entries to the directory server so that CA Single Sign-On has the necessary organization and organizational role information.

**Follow these steps:**

1. Create an LDIF file.
   **Example**: The following example contains an organization entry and an organizational role entry for the entries.ldif.

   ```
   # Netegrity, example.com
   dn: ou=Netegrity,dc= example,dc=com
   ou: Netegrity
   objectClass: organizationalUnit
   objectClass: top

   # SiteMinder, Netegrity, example.com
   dn: ou=SiteMinder,ou=CA,dc= example,dc=com
   ou: SiteMinder
   objectClass: organizationalUnit
   objectClass: top

   # PolicySvr4, SiteMinder, CA, example.com
   dn: ou=PolicySvr4,ou=SiteMinder,ou=Netegrity,dc= example,dc=com
   ou: PolicySvr4
   objectClass: organizationalUnit
   objectClass: top
   ```

```
# XPS, PolicySvr4, SiteMinder, Netegrity, example.com
dn: ou=XPS,ou=PolicySvr4,ou=SiteMinder,ou=Netegrity,dc= example,dc=com
ou: XPS
objectClass: organizationalUnit
objectClass: top
```

2. Use the following command to add the entries.

```
ldapadd -f <file_name.ldif> -D "cn=Manager,dc=example,dc=com"
-w<password>
```

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

```
Policy Store
```

4. Select the following value from the Storage list:

```
LDAP
```

5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

**Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `LDAP`
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.
2. Run the following command:

   `smreg -su password`

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).

- If the password contains a space, enclose the passphrase with quotation marks.

> **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

**Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

   - **passphrase**
     Specifies the password for the default super user account (siteminder).

     **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

   - **-adminui–setup**
     Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

   - **-t timeout**
     (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
     **Unit of measurement:** minutes

**Default:** 240 (4 hours)
**Minimum:** 15
**Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure an Oracle Directory Server as a Policy Store

Oracle Directory Server (formerly Sun Java System Directory Server) can function as a policy store. The Policy Server configuration wizard can set up this directory automatically as a policy store. However, if you did not use the wizard for set up, follow these instructions to set up the policy store up manually.

You can use this single directory instance as a policy store and key store. Using a single directory server simplifies administration tasks. This topic includes a procedure to use the policy store as a key store. If your implementation requires, you can configure a separate key store.


- Gather Directory Server Information (see page 112)
- Oracle Directory Server Enterprise Edition Considerations (see page 113)
- Point the Policy Server to the Policy Store (see page 114)
- Create the Policy Store Schema (see page 116)
- Set the CA Single Sign-On Super User Password (see page 117)
- Import the Policy Store Data Definitions (see page 118)
- Import the Default Policy Store Objects (see page 118)
- Enable the Advanced Authentication Server (see page 119)
- Restart the Policy Server (see page 120)
- Prepare for the Administrative UI Registration (see page 120)


If applicable, use the vendor–specific software to create an administrative user with the following privileges:

- create
- read
- modify
- delete

Create this user in the LDAP tree underneath the policy store root object.

### Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Oracle Directory Server Enterprise Edition Considerations

If you are using Oracle Directory Server Enterprise Edition as a policy store, consider the following.

## smldapsetup and Oracle Directory Enterprise Edition

The smldapsetup utility creates the ou=Netegrity, *root* sub suffix and PolicySvr4 database.

- **root**
  The directory root you specified in the Root DN field on the Data tab of the Policy Server Management Console. This variable has to be either an existing root suffix or sub suffix.

**Example:** If your root suffix is dc=netegrity,dc=com then running smldapsetup produces the following in the directory server:

- A root suffix, dc=netegrity,dc=com, with the corresponding userRoot database.
- A sub suffix, ou=Netegrity,dc=netegrity,dc=com, with the corresponding PolicySvr4 database.

**Example:** If you want to place the policy store under ou=apps,dc=netegrity,dc=com, then ou=apps, dc=netegrity,dc=com has to be either a root or sub suffix of the root suffix dc=netegrity,dc=com.

If it is a sub suffix, then running smldapsetup produces the following:

- A root suffix, dc=netegrity,dc=com, with the corresponding userRoot database.
- A sub suffix, ou=apps,dc=netegrity,dc=com, with the corresponding Apps database.
- A sub suffix, ou=Netegrity,ou=apps,dc=netegrity,dc=com, with the corresponding PolicySvr4 database.

**Note:** For more information about root and sub suffixes, see the Oracle documentation ( http://docs.sun.com).

## Replicate an Oracle Directory Server Enterprise Edition Policy Store

A UserRoot and a PolicySvr4 database is created. The PolicySvr4 database has suffix mappings pointing to it. To replicate this policy store, set up a replication agreement for the PolicySvr4 database directory.

**Note**: More information about a replication agreement, see the Oracle .

After you create the replication agreement, replicate the CA Single Sign-On indexes.

**Follow these steps:**

1. Generate the CA Single Sign-On indexes:

   ```
   smldapsetup ldgen -x -findexes.ldif
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

2. Set up the indexes on a replica server:

   ```
   smldapsetup ldmod -x -findexes.ldif -hhost -preplicaport
   -dAdminDN-wAdminPW
   ```

   - **host**
     Specifies the replica host.
   - **replicaport**
     Specifies the replica port number.
   - **AdminDN**
     Specifies the replica administrator DN.
     **Example:** cn=directory manager
   - **AdminPW**
     Specifies the replica administrator password.

   The CA Single Sign-On indexes are replicated.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`
4. Select the following value from the Storage list:

   LDAP
5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.
6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects. Use this procedure for the following directory server products:

- Oracle 10g, 11g (32-bit)
- Sun Java System 6.1, 7.0

Perform the following prerequisites:

- Create an LDAP instance.
- Open the Policy Server Management Console, click the Data tab, and then enter the policy store and key store information for your LDAP instance.

**Follow these steps:**

1. Run the following command from the Policy Server host system:

   ```
   smldapsetup ldgen -ffile_name
   ```

   - **file_name**
     Specifies the name of the LDIF file you are creating.

   An LDIF file with the CA Single Sign-On schema is created.

2. Run the following command:

   ```
   smldapsetup ldmod -ffile_name
   ```

   - **file_name**
     Specifies the name of the LDIF you created.

3. Run the following command:

   ```
   smldapsetup ldmod -fpolicy_server_home\xps\db\OracleDirectoryServer.ldif
   ```

   - **policy_server_home**
     Specifies the Policy Server Installation path.

4. Have the administrator of your directory server run the following command:

   ```
   dsconf reindex -h localhost -p port_number -e "ou=Netegrity,root_dn"
   ```

5. Edit the following ldif file:

   ```
   policy_server_home/xps/db/OracleDirectoryServerBrowse.ldif
   ```

6. Confirm that the LDAP directory contains the following path before proceeding (replace the Root DN below with your own Root DN):

   ```
   ou=xps,ou=PolicySvr4,ou=siteminder,ou=netegrity<Root_DN>
   ```

   Edit the following LDIF file by putting the <root dn> value from the previous step into the two places where the file has the value of <root dn>:

```
v policy_server_home/xps/db/OracleDirectoryServerBrowser.ldif
```

7. Run the following command:

```
smldapsetup ldmod -fOracleDirectoryServerBrowse.ldif -v
```

8. Stop the database and re-index the vlv indexes with the following commands:

```
dsadm stop Instance_Path
dsadm reindex -bl -t "Sort xpsSortKey" Instance_Path policysvr4
dsadm reindex -bl -t "Sort modifyTimestamp" Instance_Path policysvr4

dsadm reindex -b -t xpsNumber -t xpsValue -t xpsSortKey -t xpsCategory –
t xpsParameter -t xpsIndexedObject -t xpsTombstone instance_path policysvr4
```

9. Start the database with the following command:

```
dsadm start Instance_Path
```

The policy store schema is extended. You have created the policy store schema.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

```
smreg -su password
```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.

- The password cannot include an ampersand (&) or an asterisk (*).
- If the password contains a space, enclose the passphrase with quotation marks.

**Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

**Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure an Oracle Internet Directory Server as a Policy Store

**Contents**



Oracle Internet Directory (OID) can function as a policy store. A single directory server instance can function as a:

- Policy store

- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.
- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.
- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Index a Required Attribute

Indexing the following attribute prevents an error from occurring when you import the default policy store objects:

`modifyTimestamp`

**Follow these steps:**

1. Log in to the Oracle Internet Directory host system.
2. Use the Oracle catalog command line tool to run the following command:

   `oracle_home/ldap/bin/catalog connect=conn_str add=TRUE attribute=modifyTimestamp`

   - *oracle_home*
     Specifies the Oracle Internet Directory installation path.
   - *conn_str*
     Specifies the directory database connect string. If you have configured a tnsnames.ora file, then enter the net service name specified in the file.

   The attribute is indexed.

> **Note:** For more information about the catalog command line tool, see the Oracle documentation.

## Configure a Domain in Oracle Internet Directory

To configure an OID as a policy store, first create a domain in OID.

**Follow these steps:**

1. Open Oracle Data Manager (ODM).
2. Right-click Entry Management, and select Create.
3. Enter **dc=dcbok** for the Distinguished Name value.
4. Enter **dc** for the dc value.
5. Create an organizational unit.
6. Select an organizational unit.
7. Enter **ou=bok,dc=dcbok** for the Distinguished Name value.
8. Enter **bok** for the ou value.
   The OID domain is configured.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   > **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   ```
   Policy Store
   ```

4. Select the following value from the Storage list:

   LDAP

5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects.

**Follow these steps:**

1. Run the following command:

   `smldapsetup ldgen -f`*`file_name`*`.ldif`

   - **-f***file_name*
     Specifies the name of the schema file that you are creating.
2. Run the following command:

   `smldapsetup ldmod -f`*`file_name`*`.ldif`

   - **-f***file_name*
     Specifies the name of the schema file that you created.

3. Run the following command:

```
ldapmodify -hhost -pport -dAdminDN -wAdminPW-c -fsiteminder_home/xps/db/OID_10g.
ldif
-Z -Pcert
```

Although the schema file is version–specific, you can use this file to import the policy store schema for all supported versions of OID.

- **-h***host*
  Specifies the IP address of the LDAP directory server.
  **Example**: 123.123.12.12

- **-p***port*
  Specifies the port number of the LDAP directory server.
  **Example:** 3500

- **-d***AdminDN*
  Specifies the name of the LDAP user who has the privileges to create schema in the LDAP directory server.

- **-w***AdminPW*
  Specifies the password of the administrator specified by the -d option.

- **-c**
  Specifies continuous mode (do not stop on errors).

- **-f***siteminder_home*
  Specifies the Policy Server installation path.

- **-Z**
  (Optional) Specifies an SSL-encrypted connection.

- **-P** *cert*
  (Optional) Specifies the path of the SSL client certificate database file (cert8.db).
  **Example:**
  If cert8.db exists in app/siteminder/ssl, specify:

  ```
  -Papp/siteminder/ssl
  ```

The policy store schema is created.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

- **XPSDDInstall**
  Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```
   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```
     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).
   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```
   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure Oracle Unified Directory as a Policy Store

**Contents**



Oracle Unified Directory (OUD) can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. This scenario describes how to configure a single directory server instance to store policy data and encryption keys.

**Note**: If your implementation requires, you can configure a separate key store.

### Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## How to Configure the Oracle Unified Directory Instance

Oracle Unified Directory requires more configuration before you can use it as a policy store. The following process lists the configuration steps:

1. Specify the CA Single Sign-On schema files on Windows (see page ) or UNIX (see page ) .

2. Configure policy store indexing on Windows (see page 134) or UNIX (see page 135).

## Specify the CA Single Sign-On Schema Files on Windows

Specify the CA Single Sign-On schema files.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Navigate to *ps_home*\db\tier2\OUD and copy the following file to the schema folder ( *oud_instance*\config\schema) in the Oracle Unified Directory installation directory:
   oud_sm_schema.ldif

   - **ps_home**
     Specifies the Policy Server installation path.

   - **oud_instance**
     Specifies the name of the Oracle Unified Directory instance.

3. Navigate to *ps_home*\xps\db\schema_extension\db\OUD and copy the following file to the schema folder (*oud_instance*\config\schema) in the Oracle Unified Directory installation directory:
   oud_XPS_schema.ldif



The CA Single Sign-On schema files are specified.

## Specify the CA Single Sign-On Schema Files on UNIX

Specify the CA Single Sign-On schema files.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *ps_home*/db/tier2/OUD and copy the following file to the schema folder ( *oud_instance*/config/schema) in the Oracle Unified Directory installation directory:
   oud_sm_schema.ldif
   - **ps_home**
     Specifies the Policy Server installation path.
   - **oud_instance**
     Specifies the name of the Oracle Unified Directory instance.
3. Navigate to *ps_home*/xps/db/schema_extension/db/OUD and copy the following file to the schema folder (*oud_instance*/config/schema) in the Oracle Unified Directory installation directory:
   oud_XPS_schema.ldif

The CA Single Sign-On schema files are specified.

## Configure Policy Store Indexing on Windows

Specify indexing in the Oracle Unified Directory instance config file (*oud_instance*\config\config.ldif) to use Oracle Unified Directory as a policy store.

**Follow these steps:**

1. Stop the Oracle Unified Directory instance.
2. Open the *oud_instance*\config\config.ldif file with a text editor.
   - **oud_instance**
     Specifies the name of the Oracle Unified Directory instance.
3. Locate the following lines:

```
dn: cn=Index,cn=userRoot,cn=Workflow Elements,cn=config
objectClass: top
objectClass: ds-cfg-branch
cn: Index
```

4. Insert a new line in the file, and then add the contents from the following files:
   - *ps_home*\db/tier2\OUD\oud_sm_indexes.ldif
   - *ps_home*\xps\db\ schema_extension\db\OUD\oud_xps_index.ldif.
5. Save the file and close the text editor.
6. To rebuild the indexes, navigate to *oud_instance*\bat and run the following command .

```
rebuild-index.bat -b base_dn --rebuildAll -h hostname -
p OUD_administration_port -D "cn=Directory Manager" -j <bindpasswordfile> -X -
t 0 --completionNotify emailAddress
```

- **base_dn**

  Specifies the base DN of a back end that supports indexing. The index is rebuilt within the scope of the given base DN.

  **hostname**

  Specifies the fully qualified hostname or IP address of the directory server. If not provided, defaults to localhost.

  **OUD_administration_port**

  Specifies the administration port of the directory server. If not provided, the default administration port (44444) is used.

  **bindpasswordfile**

  Specifies the file that contains the bind password to use when authenticating to the directory server.

  **completionNotify**

  Specifies the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

7. Restart the Oracle Unified Directory instance after the rebuild task is completed.

Policy store indexing for Oracle Unified Directory is specified.

## Specify Policy Store Indexing on UNIX

Specify indexing in the Oracle Unified Directory instance config file (*oud_instance/*config/config.ldif) to use Oracle Unified Directory as a policy store.

**Follow these steps:**

1. Stop the Oracle Unified Directory instance.

2. Open the *oud_instance*/config/config.ldif file with a text editor.

   - **oud_instance**
     Specifies the name of the Oracle Unified Directory instance.

3. Locate the following lines:

   ```
   dn: cn=Index,cn=userRoot,cn=Workflow Elements,cn=config
   objectClass: top
   objectClass: ds-cfg-branch
   cn: Index
   ```

4. Insert a new line in the file, and then add the contents from the following files:

- *ps_home*/db/tier2/OUD/oud_sm_indexes.ldif
- *ps_home*/xps/db/ schema_extension/db/OUD/oud_xps_index.ldif.
- **ps_home**
  Specifies the Policy Server installation path.

5. Save the file and close the text editor.
6. Navigate to *oud_instance/*bin and run the following command to rebuild the indexes.

```
rebuild-index -b base_dn --rebuildAll -h hostname -p OUD_administration_port -
D "cn=Directory Manager" -j bindpasswordfile -X -t 0 --
completionNotify emailAddress
```

- **base_dn**

  Specifies the base DN of a back end that supports indexing. The index is rebuilt within the scope of the given base DN.

  **hostname**

  Specifies the fully qualified hostname or IP address of the directory server. If not provided, defaults to localhost.

  **OUD_administration_port**

  Specifies the administration port of the directory server. If not provided, the default administration port (44444) is used.

  **bindpasswordfile**

  Specifies the file that contains the bind password to use when authenticating to the directory server.

  **completionNotify**

  Specifies the email address of a recipient to be notified when the task completes. This option can be specified more than once in a single command.

7. Restart the Oracle Unified Directory instance after the rebuild task is completed.

The policy store indexing for Oracle Unified Directory is specified.

## How to Create the Database

The following process lists the steps for creating the directory server database for the policy store:

1. Create the base tree structure.
2. Add entries.

## Create the Base Tree Structure

To create a base tree structure to store policy store objects, specify the following entry under the root DN:

```
ou=Netegrity,ou=SiteMinder,ou=PolicySvr4
```

## Add Entries to the Database

Add entries to the directory server so that CA Single Sign-On has the necessary organization and organizational role information.

**Follow these steps:**

1. Create an LDIF file.
   **Example**: The following example contains an organization entry and an organizational role entry for the entries.ldif.

   ```
   # CA, example.com
   dn: ou=Netegrity,dc= example,dc=com
   ou: CA
   objectClass: organizationalUnit
   objectClass: top

   # SiteMinder, CA, example.com
   dn: ou=SiteMinder,ou=CA,dc= example,dc=com
   ou: SiteMinder
   objectClass: organizationalUnit
   objectClass: top

   # PolicySvr4, SiteMinder, CA, example.com
   dn: ou=PolicySvr4,ou=SiteMinder,ou=CA,dc= example,dc=com
   ou: PolicySvr4
   objectClass: organizationalUnit
   objectClass: top
   ```

2. Run the following command to add the entries.

   ```
   ldapmodify -h hostname -p port -D bindDN -j pwd_file -a -f LDIF_file_name
   ```

   - **hostname**
     Specifies the fully qualified hostname or IP address of the directory server.
     **Default**: localhost.

   - **port**
     Specifies the port on which to contact the directory server.
     **Default**: 389

   - **BindDN**
     Specifies the bind DN to authenticate to the directory server.
     **Default**: cn=Directory Manager

   - **pwd_file**
     Specifies the file that contains the bind password for authenticating to the directory server.

   - **LDIF_file_name**
     Specifies the LDIF file that you created in Step 1.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`

4. Select the following value from the Storage list:

   LDAP

5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`

9. Select the following value from the Storage list:

LDAP

10. Select the following option:

    Use Policy Store database

11. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   `smreg -su password`

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

- **npass**
  Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure Oracle Virtual Directory as a Policy Store

**Contents**



Oracle Virtual Directory (OVD) can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. This scenario describes how to configure a single directory server instance to store policy data and encryption keys.

**Note**: If your implementation requires, you can configure a separate key store.

### Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Extend the OVD Local Schema With the CA Single Sign-On Schema Files

Extend the Oracle Virtual Directory with the following CA Single Sign-On schema files:

- ovd_sm_schema.ldif
- ovd_xps_schema.ldif

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*/db/tier2/Oracle Virtual Directory and copy the following file to the OVD host system:

   ```
   ovd_sm_schema.ldif
   ```

3. Navigate to *siteminder_home*/xps/db/schema_extension/db/Oracle Virtual Directory and copy the following file to the OVD host system.

   ```
   ovd_xps_schema.ldif
   ```

4. Log in to the Oracle Virtual Directory system.
5. Have the administrator of your directory server run the following commands to extend the local schema with the CA Single Sign-On schema files:

   ```
   ldapmodify -h OVD_Host -p OVD_Port -D cn=Admin -w Admin_Password -v -a -
   f ovd_sm_schema.ldif
   ldapmodify -h OVD_Host -p OVD_Port -D cn=Admin -w Admin_Password -v -a -
   f ovd_xps_schema.ldif
   dsconf reindex -h localhost -p OVD_Port -e "ou=Netegrity,root_database"
   ```

   - **OVD_Host**
     Specifies the OVD system IP Address or fully qualified domain name.

   - **OVD_Port**
     Specifies the port on which the OVD instance is running.

   - **cn=Admin**
     Specifies the OVD server admin with rights to modify the schema.

   - **Admin_Password**
     Specifies the server admin password.

## Create an Oracle Virtual Directory Adapter to Connect to Existing Policy Store

To create an Oracle Virtual Directory LDAP adapter to connect to your existing policy store, use the Oracle Directory Services Manager.

**Follow these steps:**

1. Log in to Oracle Directory Services Manager.
2. Select Adapter from the task selection bar. The Adapter navigation tree appears.
3. Click the Create Adapter button. The New Adapter Wizard appears.
4. Specify the following values on the Type screen:
   - **Adapter Type**
     Select LDAP.
   - **Adapter Name**
     Enter a unique name for the LDAP adapter.
   - **Adapter Template**
     Select an adapter template that corresponds to the directory type of the existing policy store.
5. Click Next.
6. Enter the following values on the Connection screen (accept the defaults for all other settings):
   - **Use DNS for Auto Discovery**
     Select No.
   - **Host**
     Enter the hostname or IP address of the remote host.
   - **Port**
     Enter the port at which the remote host instance is running.
   - **Server proxy Bind DN**
     Enter the credentials of a directory user who has permission to modify directory contents.
   - **Proxy Password**
     Password for the user that is specified in the **Secure proxy Bind DN** field.
7. Click Next.
8. On the Connection Test screen, click Next if the connection status is OK. Otherwise, click Back and troubleshoot your connection settings.
9. Enter the following values on the Namespace screen (accept the defaults for all other settings):

- **Remote Base:**
  Click Browse and select the DN at which the policy data is stored.

- **Mapped Namespace**
  Enter a local DN at which to map the policy data.

10. Review the Summary page and click Finish.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.

3. Select the following value from the Database list:

   `Policy Store`

4. Select the following value from the Storage list:

   LDAP

5. Configure the following settings in the LDAP Policy Store group box.

   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.

7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

> **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  > **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure a Red Hat Directory Server as a Policy Store

**Contents**

Red Hat Directory Server can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.
- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.
- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`
4. Select the following value from the Storage list:

   LDAP
5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.
6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   smldapsetup ldgen -fschema_file
   ```

   - **schema_file**
     Specifies the name of the LDIF file you are creating.

   An LDIF file is created using the policy store schema.
3. Run the following command:

   ```
   smldapsetup ldmod -fschema_file
   ```

   - **schema_file**
     Specifies the name of the LDIF file you created.

   The policy store schema is imported.
4. Complete the following steps:

   a. Restart the directory server. Restarting the directory server is required to save the policy store schema correctly.

   b. Repeat step 3. Restarting the directory server removed the policy store root. Importing the policy store schema again is required to create the policy store root.
5. Run the following command:

   ```
   smldapsetup ldmod
   -fsiteminder_home/xps/db/RedHat_8.ldif
   ```

   - **-f***siteminder_home*
     Specifies the Policy Server installation path.

   The policy store schema is extended for XPS.

The policy store schema is created.

## Set the Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

- ***siteminder_home***
  Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

```
XPSImport fedpolicy-12.5.xml -npass
```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

**Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure Siemens DirX as a Policy Store

**Contents**



Siemens DirX can function as a policy store. A single directory server instance can function as a:

- Policy store
- Key store

Using a single directory server simplifies administration tasks. The following sections provide instruction on how to configure a single directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store.

## Gather Directory Server Information

Configuring an LDAP directory server as a policy store or upgrading an existing policy store requires specific directory server information. Gather the following information before beginning.

- **Host information**
  Specifies the fully-qualified host name or the IP Address of the directory server.
- **Port information**
  (Optional) Specifies a non-standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)
- **Administrative DN**
  Specifies the LDAP user name of a user who has privileges to create, read, modify, and delete objects in the LDAP tree underneath the policy store root object.
- **Administrative password**
  Specifies the password for the Administrative DN.
- **Policy store root DN**
  Specifies the distinguished name of the node in the LDAP tree where policy store objects are to be defined.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store and store CA Single Sign-On objects.

**Prerequisite**

Certain policy store operations require searches on CA Single Sign-On attributes. The DirX directory must index the attributes to perform search operations.

By default, the DirX directory indexes 20 attributes. CA Single Sign-On contains more attributes than this default limit. Increase the number of attributes that the directory can index to a maximum. We recommend a value of 800.

To increase the attribute indexing limit, use the following command while initializing the database profile.

```
dbamboot -Pdatabase_profile_name -aindexing_value
```

Example:

```
dbamboot -Pprofile1 -a800
```

**Create the Schema**

**Follow these steps:**

1. Create the following directory structure on the directory server host system:
   *DirX_installation*\scripts\security\CA\CA Single Sign-on

   - **DirX_installation**
     Specifies the DirX installation path.

2. Log in to the Policy Server host system.

3. Navigate to *siteminder_home*\db\tier2\SiemensDirx and copy the following files to *DirX_installation*\scripts\security\CA\CA Single Sign-on:

   - dirxabbr-ext.CA Single Sign-on*release*
   - schema_ext_for_CA Single Sign-on*release*.adm
   - bind.tcl
   - l-bind.cp
   - GlobalVar.tcl
   - **siteminder_home**
     Specifies the Policy Server installation path.

- **release**
  Specifies the CA Single Sign-On release.

4. Navigate to *siteminder_home*\xps\db\Siemens_DirX and copy the following files to *DirX_installation*\scripts\security\CA\CA Single Sign-on:
   - dirxabbr-ext.XPS
   - schema_ext_for_XPS.adm
5. Rename the following files:
   - schema_ext_for_CA Single Sign-on*release*.adm to schema_ext_for_CA Single Sign-on.adm
   - dirxabbr-ext.CA Single Sign-on*release* to dirxabbr-ext.CA Single Sign-on
   - **siteminder_home**
     Specifies the Policy Server installation path.
   - **release**
     Specifies the CA Single Sign-On release.
6. Copy the following files to *DirX_installation*\client\conf:
   - dirxabbr-ext.CA Single Sign-on
   - dirxabbr-ext.XPS
7. Restart the DirX service.
8. Go to *DirX_installation*\scripts\security\CA\CA Single Sign-on and edit the GlobalVar.tcl file to update the global variables that the DirX scripts reference.
   Default values:
   - LDAP port: 389
   - Root DN: o=My-Company
   - Admin username: cn=admin,o=My-Company
   - Admin password: dirx
9. From a command prompt, navigate to *DirX_installation*\scripts\security\CA\CA Single Sign-on and execute the following commands:
   - dirxadm schema_ext_for_siteminder.adm
   - dirxadm schema_ext_for_XPS.adm
10. Restart the DirX service.

11. On the system where DirX is installed, index the attribute modifyTimestamp. Do this task manually or through the DirX Manager.
    The steps using the DirX Manager are as follows:

    a. Rebind to the DSA.

    b. Click on Schema on the left panel.

    c. Click on Database.

    d. Uncheck "Hide attributes with no index assigned".

    e. Click Edit.

    f. Under Index row, select the attribute modifyTimestamp.

    g. Save the changes.

    h. Right click on the Database and check consistence for attribute indexes.

12. Complete the following items using the DirX manage tool:

    a. Rebind to the DSA.

    b. Create the following base tree structure:

    - Under o=My-Company, create ou=Netegrity.
    - Under ou=Netegrity, create ou=CA Single Sign-on.
    - Under ou=CA Single Sign-on, create ou=PolicySvr4
    - Under ou=PolicySvr4, create ou=XPS

The policy store schema is created.

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   `Policy Store`
4. Select the following value from the Storage list:

   LDAP
5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.
6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   LDAP
10. Select the following option:

    `Use Policy Store database`
11. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.

- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.

- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

**Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure a CA Directory Policy Store

This content describes how to configure a single CA Directory server instance to store policy data and encryption keys. If your implementation requires, you can configure a separate key store. Using a single directory server simplifies administration tasks.

- Enable the Advanced Authentication Server (see page 179)
- Prepare for the Administrative UI Registration (see page 179)

## Gather Directory Server Information

Configuring a CA Directory as a policy store requires specific directory server information. Gather the following information before configuring the policy store.

- **Host information**—Determine the fully qualified host name or the IP address of the system on which CA Directory is running.
- **DSA port number**—Determine the port on which the DSA is to listen.
- **Base DN**—Determine the distinguished name of the node in the LDAP tree in which policy store objects are to be defined.
- **Administrative DN**—Determine the LDAP user name of the account that CA Single Sign-On is to use manage objects in the DSA.
- **Administrative password**—Determine the password for the administrative user.

## Create a DSA for the Policy Store

Create the DSA by running the following command:

```
dxnewdsa DSA_Name port "o=DSA_Name,c=country_code"
```

- **DSA_Name**
  Specifies the name of the DSA.
- **port**
  Specifies the port on which the DSA is to listen.
- **o=DSA_Name,c=country_code**
  Specifies the DSA prefix.
  **Example:** "o=psdsa,c=US"

The dxnewdsa utility starts the new DSA.

**Note:** If the DSA does not automatically start, run the following:

```
dxserver start DSA_Name
```

## Create the Policy Store Schema

You create the policy store schema so the directory server can function as a policy store.

**Important!** By default, CA Directory configuration files are read–only. Any CA Directory files that you are instructed to modify, must be updated for write permission. Once the files are updated, you can revert the permission to read–only. Also, all default.xxx files provided by CA Directory are overwritten during a CA Directory upgrade. Use caution when modifying any read-only files.

**Follow these steps:**

1. Copy the following files into the CA Directory *DXHOME*\config\schema directory:
   - netegrity.dxc
   - etrust.dxc
   - **DXHOME**
     Specifies the Directory Server installation path.

     **Note:** The netegrity.dxc file is installed with the Policy Server in *siteminder_home* \eTrust. The etrust.dxc file is installed with the Policy Server in *siteminder_home* \xps\db.

   - **siteminder_home**
     Specifies the Policy Server installation path.
     - Windows %*DXHOME*%
     - Unix/Linux: $*DXHOME*
2. Create a CA Single Sign-On schema file by copying the default.dxg schema file and renaming it.
   **Note:** The default.dxg schema file is located in *DXHOME*\config\schema\default.dxg.
   **Example:** copy the default.dxg schema file and rename the copy to smdsa.dxg
3. Add the following lines to the bottom of the new CA Single Sign-On schema file:

   ```
   #CA Schema
   source "netegrity.dxc";
   source "etrust.dxc";
   ```

4. Edit the DXI file of the DSA (*DSA_Name*.dxi) by changing the schema from default.dxg to the new CA Single Sign-On schema file.
   - **DSA_Name**
     Represents the name of the DSA you created for the policy store.

     **Note:** The DXI file is located in *DXHOME*\config\servers.

5. Add the following lines to the end of the DXI file of the DSA:

```
# cache configuration
set ignore-name-bindings = true;
```

6. Copy the default limits DXC file of the DSA (default.dxc) to create a CA Single Sign-On DXC file.
**Example:** Copy the default DXC file and rename the copy smdsa.dxc.

> **Note:** The default DXC file is located in *DXHOME*\dxserver\config\limits.

7. Edit the settings in the new DXC file to match the following:

```
# size limits
set max-users = 1000;
set credits = 5;
set max-local-ops = 1000;
set max-op-size = 4000;
set multi-write-queue = 20000;
```

> **Note:** Editing the size limits settings prevents cache size errors from appearing in your CA Directory log files.

> **Important!** The multi-write-queue setting is for text–based configurations only. If the DSA is set up with DXmanager, omit this setting.

8. Save the DXC file.

9. Edit the DXI file of the DSA (*DSA_Name*.dxi) by changing the limits configuration from default.dxc to the new CA Single Sign-On limits file.
**Example:** change the limits configuration from default.dxc to smdsa.dxc.

   - **DSA_Name**
     Represents the name of the DSA you created for the policy store.

     > **Note:** The DXI file of the DSA is located in DXHOME\config\servers.If you created the DSA using DXmanager, the existing limits file is named dxmanager.dxc.

10. As the DSA user, stop and restart the DSA using the following commands:

```
dxserver stop DSA_Name
dxserver start DSA_Name
```

- **DSA_Name**
  Specifies the name of the DSA.

The policy store schema is created.

## Open the DSA

You create a view into the directory server to manage objects.

**Follow these steps:**

1. Be sure that the database is configured for an anonymous login.
2. Launch the JXplorer GUI.
3. Select the connect icon.
   Connection settings appear.
4. Enter *host_name_or_IP_address* in the Host Name field.
   - **host_name_or_IP_address**
     Specifies the host name or IP address of the system where CA Directory is running.
5. Enter *port_number* in the Port number field.
   - **port_number**
     Specifies the port on which the DSA is listening.
6. Enter o=*DSA_Name*,c=*country_code* in the Base DN field.
   **Example:** o=psdsa,c=US
7. Select Anonymous from the Level list and click Connect.
   A view into DSA appears.

## Create the Base Tree Structure for Policy Store Data

You create a base tree structure to hold policy store data. You use the JXplorer GUI to create the organizational units.

**Follow these steps:**

1. Select the root element of your DSA.
2. Create an organizational unit under the root element called:
   Netegrity
3. Create an organizational unit (root element) under Netegrity called:
   SiteMinder
4. Create an organizational unit (root element) under SiteMinder called:
   PolicySvr4

5. Create an organizational unit (root element) under PolicySvr4 called:
   XPS
   The base tree structure is created.

## Create a Superuser Administrator for the DSA

You only have to create a superuser administrator if you do not have an administrator account that CA Single Sign-On can use to access the DSA. The Policy Server requires this information to connect to the policy store.

**Follow these steps:**

1. Use the JXplorer GUI to access the DSA.
2. Create an administrator that CA Single Sign-On can use to connect to the policy store.

   **Note:** Create the user with the following object type:

   ```
   inetOrgPerson
   ```

3. Note the administrator DN and password. You use the credentials when pointing the Policy Server to the policy store.

**Example:**

```
dn:cn=admin,o=yourcompany,c=in
```

## Point the Policy Server to the Policy Store

You point the Policy Server to the policy store so the Policy Server can access the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select the following value from the Database list:

   ```
   Policy Store
   ```

4. Select the following value from the Storage list:

LDAP

5. Configure the following settings in the LDAP Policy Store group box.
   - LDAP IP Address
   - Admin Username
   - Password
   - Confirm Password
   - Root DN

   **Note**: You can click Help for a description of fields, controls, and their respective requirements.

6. Click Apply.
7. Click Test LDAP Connection to verify that the Policy Server can access the policy store.
8. Select the following value from the Database list:

   `Key Store`

9. Select the following value from the Storage list:

   LDAP

10. Select the following option:

    `Use Policy Store database`

11. Click OK.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

- *siteminder_home*
  Specifies the Policy Server installation path.

> **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

> **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Verify the CA Directory Cache Configuration

You can verify that the DXcache settings are enabled using the DXconsole.

> **Note:** By default, the DxConsole is only accessible from localhost. For more about using the set dsa command to let the DxConsole accept a connection from a remote system, see the *Directory Configuration Guide*.

**Follow these steps:**

1. From a command prompt, enter the following command to Telnet to the DSA DXConsole port:

   ```
   telnet DSA_HostDXconsole_Port
   ```

   - **DSA_Host**
     Specifies the host name or IP address of the system hosting the DSA.

**Note:** If you are on the localhost, enter **localhost**. Entering a host name or IP Address results in a failed connection.

- **DXConsole_Port**
  Specifies the port on which the DXconsole is listening. This value appears in the console-port parameter of the following file:
  *DXHOME*\config\knowledge\DSA_Name.dxc

**Default**: The DXconsole port is set to the value of the DSA port +1.**Example**: If the DSA is running on port 19389, the DXconsole port is 19390.

The DSA Management Console appears.

2. Enter the following command:

   ```
   get cache;
   ```

   The DSA Management Console displays the current DSA DXcache settings and specifies the directory caching status.

3. Enter the following command:

   ```
   logout;
   ```

   Closes the DXconsole and returns to the system prompt.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure LDAP Directory Server as Key Store

This section explains the following Key Store configurations:



## Configure a Separate Key Store

If you have a collocated policy/key store, you can configure the Policy Server to use a separate key store.

The type of directory server that is to function as a separate key store determines how you configure the store:

- If you can use the smldapsetup utility to configure a policy store, you can configure a separate key store using key store-specific schema. You can configure the following directory servers with this method:

  - Microsoft Active Directory

  - Microsoft AD LDS

- Oracle Directory Server Enterprise Edition
- Oracle Internet Directory Server
- Red Hat Directory Server

- If you cannot use the smldapsetup utility to configure a policy store, then you must:
  1. Configure a separate directory server instance with the policy store schema only. The policy store schema includes the key store schema. You do not have to:

     1. 
        - Set the super user password.
        - Import the default policy store objects.
        - Import the policy store data definitions.
          A separate key store does not require these objects.

  2. Configure the Policy Server to use this policy store instance as a key store only.

## Configure Microsoft AD LDS as a Key Store

**Contents**



You can configure Microsoft AD LDS as a separate key store.

### Key Store Prerequisites

Be sure that you meet the following prerequisites before configuring the key store:

1. Create a directory server instance that is to function as the key store. Be sure to create a root suffix and root object to store the keys.
2. Create an LDAP user with privileges to create the schema, and read, modify, and delete objects in the LDAP tree underneath the key store root object.
3. Create a key store partition.
4. Be sure that users can be created in the configuration partition. Only an administrative user in the configuration partition can import the key store schema.

## Allow User Creation in the Configuration Partition

Only an administrative user in the configuration partition can import the key store schema. This user must have administrative rights over the configuration partition and all application partitions, including the key store partition.

**Follow these steps:**

1. Open the ADSI Edit console.
2. Navigate to the following in the configuration partition:

   ```
   cn=directory service, cn=windows nt,
   cn=services, cn=configuration, cn={guid}
   ```

3. Locate the msDS-Other-Settings attribute.
4. Add the following new value to the msDS-Other-Settings attribute:

   ```
   ADAMAllowADAMSecurityPrincipalsInConfigPartition=1
   ```

5. In the configuration and policy store application partitions:
   a. Navigate to CN=Administrators, CN=Roles.
   b. Open the properties of CN=Administrators.
   c. Edit the member attribute.
   d. Click Add DN and paste the full DN of the user you created in the configuration partition.
   e. Go to the properties of the user you created and verify the value for the following object:

      ```
      msDS-UserAccountDisabled
      ```

      Be sure that the value is set false.

   The administrative user has rights over the configuration partition and all application partitions, including the key store partition.

## Gather Directory Server Information

Specific information is required to configure a separate key store. Gather the following information:

- **Host**
  The fully qualified name or the IP address of the directory server host system.
- **Port**
  The port on which the directory server instance is listening. This value is only required if the instance is listening on a non–standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrator DN**
  The full domain name, including the guid value, of the directory server administrator.
  **Example**: CN=user1,CN=People,CN=Configuration,CN,{guid}
  This user requires the following privileges:
  - create schema

    **Note:** This permission is only required to import the key store schema. After you deploy the key store, you can configure the Policy Server with a user that does not have the permission.
  - read
  - write
  - modify
  - delete
- **Administrator password**
  The password for the directory server administrator.
- **Root DN of the application partition**
  The root DN location of the application partition where the key store schema must be imported.
- **(Optional) SSL client certificate**
  If the directory connection is made over SSL, the path of the directory that contains the SSL client certificate database.

## Register the Key Store

Registering the key store configures a connection between the key store and the Policy Server. The Policy Server uses the credentials that you supply to manage the key store.

**Important!** Registration does not configure the Policy Server to use the separate key store. The settings do not take effect until the Policy Server is restarted. Do not restart the Policy Server until the key store is configured and you are ready to deploy it.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to configure the connection:

   ```
   smldapsetup reg -hhost -pport -dadmin_user -wadmin_password -rroot -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Example:**

```
smldapsetup reg -host172.16.0.0 -p389 -d"cn=directory manager" -wpassword -r"
dc=test" -k1
```

3. Start the Policy Server Management Console and open the Data tab.
4. Complete one of the following procedures:
   - If the Policy Server is configured to use a data relational database:
     a. Select Keystore from the Database list.
     b. Select LDAP from the Storage list to display the connection settings and administrative credentials.
     c. Verify that the connection settings and administrative user setting appear.
     d. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.
   - If the Policy Server is configured to use a directory server:
     a. Select Keystore from the Database list.
     b. Verify that the connection settings and the administrative user settings appear.
     c. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

   **Note:** The Use Policy Store database setting is cleared. The cleared setting is expected normal behavior. The Policy Server continues to use the key store that is collocated with the policy store.

5. Exit the Policy Server Management Console.
   The separate key is registered with the Policy Server.

## Create the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to create the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to create the key store schema file:

   ```
   smldapsetup ldgen -ffile_name -k1
   ```

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Note:** For more information about these modes and arguments, see the *Policy Server Administration Guide*.
**Example:** smldapsetup ldgen -fkeystoreschema -k1
The key store schema file is created.

### Import the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to import the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to import the key store schema:

   ```
   smldapsetup ldmod -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   Consider the following items:

   - For more information about these modes and arguments, see the *Policy Server Administration Guide*.
   - Standard out displays all policy store schema being imported. The behavior is normal and expected. The utility only imports the key–store specific schema.

   **Example:** smldapsetup ldmod -fkeystoreschema -k1
   The key store–specific schema is imported.

### Restart the Policy Server

The Policy Server continues to use the collocated key store until you restart the Policy Server. Restart the Policy Server to begin using the separate key store.

**Note:** For more information, see the *Policy Server Administration Guide*.

## Configure Microsoft Active Directory as a Key Store

**Contents**

You can configure Microsoft Active Directory as a separate key store.

## Key Store Prerequisites

1. Create a directory server instance that is to function as the key store. Be sure to create a root suffix and root object to store the keys.

2. Create an LDAP user with privileges to create the schema, and read, modify, and delete objects in the LDAP tree underneath the key store root object.

## Gather Directory Server Information

Specific information is required to configure a separate key store. Gather the following information:

- **Host**
  The fully qualified name or the IP Address of the directory server host system.

- **Port**
  The port on which the directory server instance is listening. This value is only required if the instance is listening on a non–standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrative DN**
  Specifies the LDAP user name of a user that has privileges to:

  - create schema

  **Note:** This permission is only required to import the key store schema. After you deploy the key store, you can configure the Policy Server with a user that does not have the permission.

  - read
  - write
  - modify
  - delete

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Key store root DN**
  Specifies the distinguished name of the node in the LDAP tree where the key store objects must be imported.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Register the Key Store

Registering the key store configures a connection between the key store and the Policy Server. The Policy Server uses the credentials that you supply to manage the key store.

**Important!** Registration does not configure the Policy Server to use the separate key store. The settings do not take effect until the Policy Server is restarted. Do not restart the Policy Server until the key store is configured and you are ready to deploy it.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to configure the connection:

   ```
   smldapsetup reg -hhost -pport -dadmin_user -wadmin_password -rroot -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **Example:**

   ```
   smldapsetup reg -host172.16.0.0 -p389 -d"cn=directory manager" -wpassword -r"
   dc=test" -k1
   ```

3. Start the Policy Server Management Console and open the Data tab.
4. Complete one of the following procedures:
   - If the Policy Server is configured to use a data relational database:
     a. Select Keystore from the Database list.
     b. Select LDAP from the Storage list to display the connection settings and administrative credentials.
     c. Verify that the connection settings and administrative user setting appear.
     d. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

- If the Policy Server is configured to use a directory server:
  a. Select Keystore from the Database list.
  b. Verify that the connection settings and the administrative user settings appear.
  c. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

**Note:** The Use Policy Store database setting is cleared. The cleared setting is expected normal behavior. The Policy Server continues to use the key store that is collocated with the policy store.

5. Exit the Policy Server Management Console.
   The separate key is registered with the Policy Server.

## Create the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to create the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to create the key store schema file:

   ```
   smldapsetup ldgen -ffile_name -k1
   ```

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Note:** For more information about these modes and arguments, see the *Policy Server Administration Guide*.
**Example:** smldapsetup ldgen -fkeystoreschema -k1
The key store schema file is created.

## Import the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to import the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to import the key store schema:

   ```
   smldapsetup ldmod -ffile_name -k1
   ```

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

Consider the following items:

- For more information about these modes and arguments, see the *Policy Server Administration Guide*.
- Standard out displays all policy store schema being imported. The behavior is normal and expected. The utility only imports the key–store specific schema.

**Example:** smldapsetup ldmod -fkeystoreschema -k1
The key store–specific schema is imported.

### Restart the Policy Server

The Policy Server continues to use the collocated key store until you restart the Policy Server. Restart the Policy Server to begin using the separate key store.

**Note:** For more information, see the *Policy Server Administration Guide*.

## Configure an Oracle Internet Directory Server as a Key Store

**Contents**



You can configure Oracle Internet Directory Server as a separate key store.

### Key Store Prerequisites

1. Create a directory server instance that is to function as the key store. Be sure to create a root suffix and root object to store the CA Single Sign-On keys.
2. Create an LDAP user with privileges to create the schema, and read, modify, and delete objects in the LDAP tree underneath the key store root object

### Gather Directory Server Information

Specific information is required to configure a separate key store. Gather the following information:

- **Host**
  The fully qualified name or the IP Address of the directory server host system.

- **Port**
  The port on which the directory server instance is listening. This value is only required if the instance is listening on a non–standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrative DN**
  Specifies the LDAP user name of a user that has privileges to:

  - create schema

  **Note:** This permission is only required to import the key store schema. After you deploy the key store, you can configure the Policy Server with a user that does not have the permission.

  - read
  - write
  - modify
  - delete

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Key store root DN**
  Specifies the distinguished name of the node in the LDAP tree where the key store objects must be imported.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Register the Key Store

Registering the key store configures a connection between the key store and the Policy Server. The Policy Server uses the credentials that you supply to manage the key store.

**Important!** Registration does not configure the Policy Server to use the separate key store. The settings do not take effect until the Policy Server is restarted. Do not restart the Policy Server until the key store is configured and you are ready to deploy it.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to configure the connection:

```
smldapsetup reg -hhost -pport -dadmin_user -wadmin_password -rroot -k1
```

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Example:**

```
smldapsetup reg -host172.16.0.0 -p389 -d"cn=directory manager" -wpassword -r"
dc=test" -k1
```

3. Start the Policy Server Management Console and open the Data tab.
4. Complete one of the following procedures:
   - If the Policy Server is configured to use a data relational database:
     a. Select Keystore from the Database list.
     b. Select LDAP from the Storage list to display the connection settings and administrative credentials.
     c. Verify that the connection settings and administrative user setting appear.
     d. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.
   - If the Policy Server is configured to use a directory server:
     a. Select Keystore from the Database list.
     b. Verify that the connection settings and the administrative user settings appear.
     c. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

   **Note:** The Use Policy Store database setting is cleared. The cleared setting is expected normal behavior. The Policy Server continues to use the key store that is collocated with the policy store.

5. Exit the Policy Server Management Console.
   The separate key is registered with the Policy Server.

## Create the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to create the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to create the key store schema file:

   ```
   smldapsetup ldgen -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **Note:** For more information about these modes and arguments, see the *Policy Server Administration Guide*.
   **Example:** smldapsetup ldgen -fkeystoreschema -k1
   The key store schema file is created.

## Import the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to import the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to import the key store schema:

   ```
   smldapsetup ldmod -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   Consider the following items:

   - For more information about these modes and arguments, see the *Policy Server Administration Guide*.
   - Standard out displays all policy store schema being imported. The behavior is normal and expected. The utility only imports the key–store specific schema.

   **Example:** smldapsetup ldmod -fkeystoreschema -k1
   The key store–specific schema is imported.

## Restart the Policy Server

The Policy Server continues to use the collocated key store until you restart the Policy Server. Restart the Policy Server to begin using the separate key store.

**Note:** For more information, see the *Policy Server Administration Guide*.

# Configure a Red Hat Directory Server as a Key Store

**Contents**



You can configure Red Hat Directory Server as a separate key store.

## Key Store Prerequisites

1. Create a directory server instance that is to function as the key store. Be sure to create a root suffix and root object to store the keys.

2. Create an LDAP user with privileges to create the schema, and read, modify, and delete objects in the LDAP tree underneath the key store root object.

## Register the Key Store

Registering the key store configures a connection between the key store and the Policy Server. The Policy Server uses the credentials that you supply to manage the key store.

**Important!** Registration does not configure the Policy Server to use the separate key store. The settings do not take effect until the Policy Server is restarted. Do not restart the Policy Server until the key store is configured and you are ready to deploy it.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Run the following command to configure the connection:

   ```
   smldapsetup reg -hhost -pport -dadmin_user -wadmin_password -rroot -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **Example:**

   ```
   smldapsetup reg -host172.16.0.0 -p389 -d"cn=directory manager" -wpassword -r"
   dc=test" -k1
   ```

3. Start the Policy Server Management Console and open the Data tab.

4. Complete one of the following procedures:

   - If the Policy Server is configured to use a data relational database:

     a. Select Keystore from the Database list.

     b. Select LDAP from the Storage list to display the connection settings and administrative credentials.

     c. Verify that the connection settings and administrative user setting appear.

     d. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

   - If the Policy Server is configured to use a directory server:
     a. Select Keystore from the Database list.
     b. Verify that the connection settings and the administrative user settings appear.
     c. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

   **Note:** The Use Policy Store database setting is cleared. The cleared setting is expected normal behavior. The Policy Server continues to use the key store that is collocated with the policy store.

5. Exit the Policy Server Management Console.
   The separate key is registered with the Policy Server.

## Create the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to create the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Run the following command to create the key store schema file:

   ```
   smldapsetup ldgen -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Note:** For more information about these modes and arguments, see the *Policy Server Administration Guide*.
**Example:** smldapsetup ldgen -fkeystoreschema -k1
The key store schema file is created.

### Import the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to import the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to import the key store schema:

   ```
   smldapsetup ldmod -ffile_name -k1
   ```

   **Note:** Standard out displays all policy store schema being imported. The behavior is normal and expected. The utility only imports the key–store specific schema. The utility only imports the key–store specific schema.

   **Example:**

   ```
   smldapsetup ldmod -fkeystoreschema -k1
   ```

   The key store–specific schema is imported.
3. Complete the following steps:

   a. Restart the directory server. Restarting the directory server is required to save the key store schema correctly.

   b. Repeat step 2. Restarting the directory server removed the key store root. Importing the key store schema again is required to create the key store root.

   The key store–specific schema is imported.

### Restart the Policy Server

The Policy Server continues to use the collocated key store until you restart the Policy Server. Restart the Policy Server to begin using the separate key store.

**Note:** For more information, see the *Policy Server Administration Guide*.

## Configure an Oracle Directory Server Enterprise Edition as a Key Store

**Contents**

You can configure Oracle Directory Server Enterprise Edition as a separate key store.

## Key Store Prerequisites

1. Create a directory server instance that is to function as the key store. Be sure to create a root suffix and root object to store the keys.
2. Create an LDAP user with privileges to create the schema, and read, modify, and delete objects in the LDAP tree underneath the key store root object.

## Key Store Considerations

The smldapsetup utility creates the ou=Netegrity, *root* sub suffix and PolicySvr4 database.

- **root**
  The directory root you specify when registering the key store. This variable has to be either an existing root suffix or sub suffix.

**Example:** If your root suffix is dc=netegrity,dc=com then running smldapsetup produces the following entries in the directory server:

- A root suffix, dc=netegrity,dc=com, with the corresponding userRoot database.
- A sub suffix, ou=Netegrity,dc=netegrity,dc=com, with the corresponding PolicySvr4 database.

If you want to place the key store under ou=apps,dc=netegrity,dc=com, then ou=apps,dc=netegrity, dc=com has to be either a root or sub suffix of the root suffix dc=netegrity,dc=com.

If it is a sub suffix, then running smldapsetup produces the following entries:

- A root suffix, dc=netegrity,dc=com, with the corresponding userRoot database.
- A sub suffix, ou=apps,dc=netegrity,dc=com, with the corresponding Apps database.
- A sub suffix, ou=Netegrity,ou=apps,dc=netegrity,dc=com, with the corresponding PolicySvr4 database.

**Note:** For more information about root and sub suffixes, see your vendor–specific documentation.

## Gather Directory Server Information

Specific information is required to configure a separate key store. Gather the following information:

- **Host**
  The fully qualified name or the IP Address of the directory server host system.

- **Port**
  The port on which the directory server instance is listening. This value is only required if the instance is listening on a non–standard port.
  **Default values:** 636 (SSL) and 389 (non-SSL)

- **Administrative DN**
  Specifies the LDAP user name of a user that has privileges to:

  - create schema

  **Note:** This permission is only required to import the key store schema. After you deploy the key store, you can configure the Policy Server with a user that does not have the permission.

  - read
  - write
  - modify
  - delete

- **Administrative password**
  Specifies the password for the Administrative DN.

- **Key store root DN**
  Specifies the distinguished name of the node in the LDAP tree where the key store objects must be imported.

- **SSL client certificate**
  Specifies the pathname of the directory where the SSL client certificate database file resides.
  **Limit:** SSL only

## Register the Key Store

Registering the key store configures a connection between the key store and the Policy Server. The Policy Server uses the credentials that you supply to manage the key store.

**Important!** Registration does not configure the Policy Server to use the separate key store. The settings do not take effect until the Policy Server is restarted. Do not restart the Policy Server until the key store is configured and you are ready to deploy it.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to configure the connection:

```
smldapsetup reg -hhost -pport -dadmin_user -wadmin_password -rroot -k1
```

> **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Example:**

```
smldapsetup reg -host172.16.0.0 -p389 -d"cn=directory manager" -wpassword -r"
dc=test" -k1
```

3. Start the Policy Server Management Console and open the Data tab.
4. Complete one of the following procedures:
   - If the Policy Server is configured to use a data relational database:
     a. Select Keystore from the Database list.
     b. Select LDAP from the Storage list to display the connection settings and administrative credentials.
     c. Verify that the connection settings and administrative user setting appear.
     d. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.
   - If the Policy Server is configured to use a directory server:
     a. Select Keystore from the Database list.
     b. Verify that the connection settings and the administrative user settings appear.
     c. Click test LDAP Connection to verify that the Policy Server can communicate with the key store instance.

> **Note:** The Use Policy Store database setting is cleared. The cleared setting is expected normal behavior. The Policy Server continues to use the key store that is collocated with the policy store.

5. Exit the Policy Server Management Console.
   The separate key is registered with the Policy Server.

## Create the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to create the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to create the key store schema file:

   ```
   smldapsetup ldgen -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **Note:** For more information about these modes and arguments, see the *Policy Server Administration Guide*.
   **Example:** smldapsetup ldgen -fkeystoreschema -k1
   The key store schema file is created.

## Import the Key Store Schema

The key store instance requires the schema to store and retrieve CA Single Sign-On web agent keys. Use the smldapsetup utility to import the key store schema file.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command to import the key store schema:

   ```
   smldapsetup ldmod -ffile_name -k1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   Consider the following items:

   - For more information about these modes and arguments, see the *Policy Server Administration Guide*.
   - Standard out displays all policy store schema being imported. The behavior is normal and expected. The utility only imports the key–store specific schema.

**Example:** smldapsetup ldmod -fkeystoreschema -k1
The key store–specific schema is imported.

## Restart the Policy Server

The Policy Server continues to use the collocated key store until you restart the Policy Server. Restart the Policy Server to begin using the separate key store.

**Note:** For more information, see the *Policy Server Administration Guide*.

## Replicate an Oracle Directory Server Key Store

CA Single Sign-On creates a UserRoot and a PolicySvr4 database. Suffix mappings point to the PolicySvr4 database. Replicating a key store requires that you set up a replication agreement for the PolicySvr4 database directory.

**Follow these steps:**

1. Configure a replication agreement as detailed by your vendor–specific documentation.
2. Log in to the Policy Server host system.
3. Run the following command to generate the CA Single Sign-On indexes:

   ```
   smldapsetup ldgen -x -findexes.ldif
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

4. Set up the indexes on a replica server:

   ```
   smldapsetup ldmod -x -findexes.ldif -hhost -preplicaport
   -dAdminDN-wAdminPW
   ```

   - **host**
     Specifies the replica host.
   - **replicaport**
     Specifies the replica port number.
   - **AdminDN**
     Specifies the replica administrator DN.
     **Example:** cn=directory manager
   - **AdminPW**
     Specifies the replica administrator password.

   The CA Single Sign-On indexes are replicated.

# Configure CA Directory as a Session Store

You can configure CA Directory as a session store by completing the following procedures:



The procedures assume that CA Directory is installed and operating.

In this topic use the following convention:

- DXHOME= C:\Program Files\CA\Directory\dxserver (Windows)
- DXHOME=/opt/CA/Directory/dxserver (UNIX)

## Locate the Session Store Schema File

In the session store, the Policy Server uses its own LDAP directory objects. CA Directory has to know about the structure of those extra classes. The session store schema file, netegrity.dxc, lets CA Directory know the names and structure of those extra classes.

The **netegrity.dxc** file is installed with the Policy Server. Verify that you have the correct schema file.

**Follow these steps:**

1. Navigate to the directory *siteminder_home*\eTrust, where *siteminder_home* specifies the Policy Server installation path
2. Verify that you have the schema file **netegrity.dxc**. This file creates the DSA session store schema, which lets the DSA store and retrieve user session information. If you do not have this file, contact your CA Single Sign-On Administrator and request it.

## Create a Directory System Agent (DSA) for the Session Store

Create a DSA and dedicate its use only to the session store. A dedicated DSA helps to maximize session store performance.

**Follow these steps:**

1. Log in to the CA Directory host system.
2. Create a data DSA by running the following command:

   `dxnewdsa dsa_name port prefix`

   - *dsa_name* specifies the name of the session store DSA.

- *port* specifies the port on which the session store must listen for requests.
- *prefix* specifies the namespace prefix. Use LDAP syntax to specify the prefix.

Example:

```
dxnewdsa smsessionstore 1234 o=forwardinc,c=us
```

Forwardinc is a fictitious company name. The name is used strictly for instructional purposes only and is not meant to reference an existing company.

The DSA is created.

## Create the Session Store Schema

The DSA requires the schema to store and retrieve the user session information.

**Follow these steps:**

1. Log in to the CA Directory host system.
2. Stop the DSA using the following command:

   ```
   dxserver stop DSA_name
   ```

   *DSA_name* is the name that you assign to the DSA.
3. Copy the **netegrity.dxc** file from *siteminder_home*\eTrust.
4. Navigate to *DXHOME*\config\schema and paste **netegrity.dxc** into the directory.
5. Create a schema file by following these steps:
   a. Copy the default DSA schema file (default.dxg). The default.dxg file is in the schema directory.
   b. Remove the read-only attribute from the copy.
   c. Rename the copy to create a new file. For example, rename the copy to smsession.dxg.
6. Add the following lines to the bottom of the new schema file:

   ```
   #CA Schema
   source "netegrity.dxc";
   ```
7. Save the file.
8. Reapply the read-only attribute to the new schema file.

## Set the Session Store Limits on the DSA

In the session store schema, set the session store default limits. The limits define DSA operations, such as how long an application stays connected to the DSA or how search queries are controlled.

**Follow these steps:**

1. Navigate to *DXHOME*\config\limits.
2. Create a session store limits file by complete these steps:
    a. Copy the default limits file (default.dxc).
    b. Remove the read-only attribute.
    c. Rename the file, such as smsession.dxc
3. In the size limits section of the **smsession.dxc** file, set the **max-ops** setting to match the following example. This value represents a high limit. The session store is not expected to return more than 1,000 objects per search query.

    ```
    set max-op-size = 1000;
    ```

4. Configure the **multi-write queue** setting. The multiwrite queue setting specifies the maximum number of transactions that are held in memory for a DSA that is unavailable. Use this setting to support a server that is offline for a given period. The default value is 20,000. Set the queue to handle from 300,000 through 500,000 transactions. The optimal value allows a normal reboot of one of the directory server hosts without causing the multiwrite queue to fill. For example:

    ```
    set multi-write-queue = 300000;
    ```

5. Set the **multi-write-outstanding-ops limits** setting. This setting determines the number of updates that a DSA can send at one time, from its multiwrite queue to a recovering DSA. The default is 10. Based on the number of potential updates in the multiwrite queue, a DSA sending the default 10 updates at a time prolongs the automatic recovery of a DSA. The recommenced value is 1000. For example:

    ```
    set multi-write-outstanding-ops = 1000;
    ```

6. Save the file.
7. Reapply the read-only attribute.

## Modify the Session Store Initialization File

The initialization file (*dsa_name*.dxi) contains settings for configuring the following functions:

- Indexes
  Optimization of the CA Directory indexes is critical for the overall performance of CA Directory. If you do not properly configure the indexes, the directory might experience slower response times for add and remove operations. Specify the attributes that require indexing in the **set cache-index** entry.
- Storing session objects in memory
- Transaction logging
  By default, each CA Directory Data DSA is configured to use a transaction log. The session store does not need the transaction log and it can negatively affect the performance of the directory. So, disable logging.

**Follow these steps:**

1. Navigate to *DXHOME*\config\servers and open the session store initialization file, named ***DSA_name*.dxi**. *DSA_name* specifies the name of the session store DSA.

2. In the #schema section, edit the schema reference from default.dxg to the new file. **Example:** Change default.dxg to smsession.dxg.

3. In the #service limits section, edit the service limits reference from default.dxc to the new file. **Example:** Change default.dxc to smsession.dxc.

4. In the #grid configuration section, edit the **set cache-index** entry to match the following text:

   ```
   set cache-index = smSessionId, smExpirationTime, smIdleExpirationTime,
   smSearchData, smVariableName, smFullVariableName;
   ```

   Verify that the **set cache-index** entry is above the **set lookup-cache** entry, for example:

   ```
   set cache-index = smSessionId, smExpirationTime, smIdleExpirationTime,
   smSearchData, smVariableName, smFullVariableName;
   set lookup-cache = true;
   ```

5. (Optional) To store more session objects in memory, compress the smVariableValue attribute using the following command:

   ```
   set compress type=smVariableValue
   ```

   Where *type* is one of the following values: base64, hex, deflate, deflate1, deflate2. The deflate type is fastest, deflate1 balances speed with compression, and deflate2 tries for maximum compression.

   Use only base64 when all the values of the specified attributes are base64 encoded. Specify the hex type only when all values of the specified attributes are hex encoded. If the values of an attribute are encoded using any type other than base64 or hex, then use one of the deflate types.

6. Ensure that the **multi-write-disp-recovery** setting is set to false, the default setting. This session store does not need this recovery method. The correct entry is shown in the following example:

   ```
   set multi-write-disp-recovery = false;
   ```

   Routing DSAs do not need this setting.

   You can set up multiple DSA peers and use multiwrite replication, but *do not* use this feature with DISP recovery. For instructions on setting up replication ( https://docops.ca.com/display/CAD1217/CA+Directory), see the CA Directory documentation.

7. (Optional) Disable transaction logging to improve performance. It is enabled by default. To disable logging, add the following setting to each initialization file:

   ```
   set disable-transaction-log = true;
   ```

Place the logging entry *below* the entry set multi-write-disp-recovery = false. For example:

```
set multi-write-disp-recovery = false;
set disable-transaction-log = true;
```

Disabling the transaction logging is not required for the routing DSAs. Consider the effects disabling transaction logging has on data recovery. For more information, see the CA Directory documentation.

8. (Optional) If session store DSAs are replicated, add the dsp-idle-time setting to the DSA **knowledge** file (*DXHOME*/config/knowledge/). Setting the idle time is useful so that the idle time is not too low. If set too low, the link between a multiwrite DSA and its recovering peer DSA can time out before the DSA is recovered.

   ```
   dsp-idle-time = 30
   ```

9. Restart the DSA using the following command:

   ```
   dxserver start DSA_Name
   ```

The session store schema is configured.

## Session Store Backups Not Required

It is not necessary to to back up the session store as the data within it is always changing. Restored data would therefore be out-of-date and inaccurate.

## Asynchronous Replication

You can configure DSA replication so that the same directory information is stored on several servers. Replication is deployed to improve the availability of directory servers so applications can continue working. Replication can also improve performance by enabling load balancing between DSAs.

The CA Directory session store processes update requests (modify/add/remove) at a high volume. To optimize the replication configuration across all the DSA session stores including the local stores, set replication to asynchronous. Updates that are sent asynchronously free up threads so that they can continue processing requests without waiting for validation. Therefore, replication performance is efficient.

**Follow these steps:**

1. Navigate to *DXHOME*/config/knowledge/ of the DSA.

2. Set the option **multi-write-async** as a dsa-flag in the knowledge configuration of all DSAs. For example:

   ```
   [ dsa-flags  = multi-write-async, no-service-while-recovering ]
   ```

## Enable the Policy Server to Manage the Session Store

For the Policy Server to manage the session store, complete the following tasks:

- Add a session store administrative user to the DSA
- Establish a root DN for the session store.

- Point the Policy Server to the session store.

## Add a Session Store Administrative User and Root DN for the DSA

For the Policy Server to manage the session store, it requires the following information:

- The complete distinguished name (DN) and password of a user in the DSA. The Policy Server uses these credentials to manage the session store.
- A root DN to which session information can be written.

**Follow these steps:**

1. Access the DSA using anonymous authentication with *one* of the following methods:
   - Use the JXplorer tool.
   - Use the CA Directory command-line interface (https://support.ca.com/cadocs/0/CA%20Directory%20r12%200%20SP11-ENU/Bookshelf_Files/HTML/idocs/dxmodify_tool.htm).

   These procedures use the JXplorer tool.
2. Create a user that the Policy Store can use to manage the session store.
   - Be sure to create the user with only the following OBJECT CLASS: **inetOrgPerson**
   - Note the credentials. The credentials are required to point the Policy Server to the session store DSA.
3. Disconnect from JXplorer.
4. Start JXplorer.
5. Log in to the DSA using the complete DN of the administrative user you created to verify that you can access the DSA.
   **Example:** cn=admin,o=forwardinc,c=us
6. Manually create an organizational unit that serves as the root DN of the session store.
   **Example:** ou=sessionstore
7. Disconnect from JXplorer.

**Note:** We recommend that you disable the anonymous authentication to prevent unauthorized access to the session store.

## Point the Policy Server to the Session Store

Point the Policy Server to the session store DSA so the Policy Server can manage the session store.

**Follow these steps:**

1. Open the Policy Server Management Console.

**Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the release notes for your CA Single Sign-On component.

2. Click the Data tab.
3. Select Session Store from the Database list.
4. Select CA Directory from the Storage list.
5. Select the Session Store Enabled option.
6. Under LDAP Session Store section:
    a. Enter the IP address and port of the session store DSA.
    b. Enter the root DN of the session store DSA.
    **Example:** ou=sessionstore,o=fowardinc,c=us
    c. Enter the complete DN of an administrative user in the DSA.
    **Example:** cn=admin,o=forwardinc,c=us
    d. Enter the password of the administrative user.
7. Click Test LDAP Connection to verify the connection.
8. Click OK.

The Policy Server is configured to manage the session store.

## Session Store Performance Optimization

If your environment requires high performance, consider the following recommendations:

- CA Directory supports namespace partitioning which allows for the data for a single OU to be divided up among multiple DSAs. This can have a significant impact on performance by allowing more DSAs to handle the LDAP request load. Data Partitioning allows you to take advantage of all available hardware, including extra CPUs on a single host or CPUs across multiple hosts. The more replicas you have, the more parallel writes to the directory can occur.

  If your environment has CPUs or cores on the existing host or on other hosts that not being leveraged, consider configuring partitioning the CA Directory namespace.

- Do not index all session store attributes. Configuring a high number of indexes on session data degrades deletion performance. Only index session store attributes that are used in your deployment. To determine which indexes are used in a deployment, see the CA Directory documentation.

- Use partitions but have each partition on a separate host.

- Allow enough physical memory to contain all CA Directory server data files for each host. The host can be a physical system or a guest instance. You can view the size of the data files in the data directory of the CA Directory server installation.

- Verify the following entries in the initialization file *(dsa_name*.dxi) for your DSA. This file resides in the directory *DXHome*/config/servers. If these entries are not in the file, add them.

  - Disable CA Directory transaction logging by adding the following command to the server file:

    ```
    # disable transaction logging for performance
    set disable-transaction-log = true;
    set disable-transaction-log-flush = true;
    ```

    **Note:** If there is a service interruption, disabling transaction logging without enabling replication requires users to log in again.

  - Use a default single queue by adding the dxgrid-queue command. The setting for this command defaults to "true." For example:

    ```
    # use single queue in front of DSA instead of
    # one queue per thread
    set dxgrid-queue = true;
    ```

## Connection Pools

A connection pool is a cache of database connection handles that is maintained so that the connections can be reused when future requests to the database are required. The Policy Server uses a connection pool (of 10 LDAP connection handles by default) to enhance session store performance.

However, if too many concurrent threads require a session store connection handle, the connection pool can be overwhelmed and the thread queue can build up. To diagnose this issue, search the Policy Server profiler log for requests with a delay before LDAP handles are locked. Such a delay indicates that you must increase the number of session store connections

For example, the following excerpt from the profiler log shows a two-second wait before the worker thread ID 3003599728 was able to get an LDAP connection handle:

```
[13:23:38.671][26918][3003599728][Leave function CSm_Auth_Message::SetAuthContext]
[Sm_Auth_Message.cpp:4280][CSm_Auth_Message::SetAuthContext]
[13:23:40.671][26918][3003599728][Lock LDAP handle. slot=0 ld=0xb3071fe0][LdapStore.
cpp:375][Lock_LdapHandle]
```

(The delay is indicated by the 2-second difference between the message timestamps, which are shown in bold.)

To increase the size of the session store connection pool, modify the value MaxConnections registry setting, which can be found at the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder\CurrentVersion\LdapSessionServer=numeric_id
```

**Value**: Connection pool size in hex. For example, 0xf (15)

**Default**: 0xa (10)

**Note**: Configure a value close to half the max worker thread value so you have at least one connection handle for every 2 worker threads. This recommendation varies based on the hardware that is used and other environmental factors. For optimal performance, we recommend testing these settings in a test environment and modify as required to adjust to your environment.

### Deleting Excess Expired Sessions in High Volume Environments

Policy Servers that are configured to use CA Directory as a session store are responsible for creating sessions and then cleaning up the expired sessions. The Policy Servers create sessions when users authenticate, then periodically, the Policy Server searches for sessions that have expired and removes them.

In high volume environments, the CA Directory session store can fill with excess expired sessions. If you encounter this issue, open an issue with CA Support, who can provide you with the SMDeleteSession utility, which continuously checks for and removes expired sessions.

# Configure ODBC Databases as Policy, Session, Key and Audit Stores

The content in this section describes how to configure an ODBC data stores, which include:

- Policy Store (see page 210)
- Session Store (see page 212)
- Key Store (see page 212)
- Audit Store (see page 213)

# Policy Store

The CA Single Sign-On policy store is the repository for all policy-related information. All Policy Servers in a CA Single Sign-On installation must share the policy store data, either directly or through replication. CA Single Sign-On is installed with tools that let administrators move policy store data from one storage facility to another.

When you install the Policy Server, you can automatically configure one of the following relational databases as a policy store:

- Microsoft SQL Server
- PostgreSQLServer
- Oracle RDBMS

Optionally, you can manually configure a policy store after installing the Policy Server. After you install the Policy Server, you can also use the Policy Server Management Console to point the Policy Server to an existing policy store. For a list of supported CA and third-party components, refer to the Platform Support Matrix on the Technical Support site.

In addition to policy store support, you can use a relational database to store keys, audit logs, and session data.

**Important!** Consider the following issues before configuring a policy store:

- Avoid possible policy store corruption. Be sure that the database server that is to host the policy store is configured to store objects in UTF-8 form:
  - (Oracle) Be sure that the database is configured to store objects in UTF-8 form. Oracle supports unicode within many of their character sets.
  - (SQL Server) Be sure that the database is configured using the default collation (SQL_Latin1_General_CP1_CI_AS). Using a collation that is case-sensitive can result in unexpected behaviors.
- Do not use brackets around the IP address when using IPv6 ODBC data sources or the connection fails. For example, use **fec0::9255:20c:29ff:fe47:8089** not [fec0::9255:20c:29ff:fe47:8089].

## Default Policy Store Objects Consideration

When you configure a policy store, the following default policy store object files are available:

- smpolicy.xml
- smpolicy-secure.xml

Both files contain the default objects that the policy store requires.

If you use the Policy Server Configuration Wizard to configure the policy store automatically, the wizard only uses smpolicy.xml. If you want to use smpolicy–secure.xml, configure the policy store manually.

Both files provide default security settings. These settings are available in the default Agent Configuration Object (ACO) templates that are available in the Administrative UI. The smpolicy-secure file provides more restrictive default security settings. Choosing smpolicy.xml does not limit you from using the more restrictive default security settings. You can modify the default ACO settings using the Administrative UI.

The following table summarizes the security settings for both files:

| Parameter Name | smpolicy Values | smpolicy–secure Values |
|---|---|---|
| BadCssChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadQueryChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadUrlChars | //, ./, /., /*, *., ~, \, %00-%1f, %7f-%ff, %25 | smpolicy.smdif values plus:<br><, >, ', ;, ), (, &, + |
| EnableCookieProvider | Yes | No |
| IgnoreExt | .class, .gif, .jpg, .jpeg, .png, .fcc, .scc, .sfcc, .ccc, .ntc | All smpolicy values. |
| LimitCookieProvider | No | Yes |
| ValidTargetDomain | This file does not include this parameter. | This parameter does not have a default value. Provide a valid redirection domain.<br>Example: validtargetdomain=".example.com (http://example.com)" |

# Session Store

The session store is where the Policy Server stores persistent session data. A persistent session is one in which a cookie is maintained in the session store, in the memory of the web browser, and optionally the hard disk.

Before you implement persistent sessions, consider the following information:

- Persistent sessions are configured on a per realm basis.
- Use Persistent sessions only when necessary. Using session services to maintain sessions has an impact on system performance.

If you plan to use persistent sessions in one or more realms, enable the session store using the Policy Server Management Console.

# Key Store

The key store holds web agent keys and session ticket keys, which are distributed to Agents at run time.

Web Agents use an agent key to encrypt cookies before passing the cookies to a browser. When a Web Agent receives a CA Single Sign-On cookie, the agent key enables the Agent to decrypt the contents of the cookie. Keys must be set to the same value for all Web Agents communicating with a Policy Server.

The Policy Server and Agents use encryption keys to encrypt and decrypt sensitive data that is passed between Policy Servers and Agents.

- The Agent uses agent keys to encrypt CA Single Sign-On cookies that are read and shared by all agents in a single sign-on environment. The agent key also decrypts cookies encrypted by the other agents. The Policy Server manages agent keys and distributes the keys to agents periodically.

- Session ticket keys are used by the Policy Server to encrypt session tickets. Session tickets contain credentials and other information relating to a session (including user credentials). Agents embed session tickets in CA Single Sign-On cookies, but cannot do not have access to the session ticket keys, which never leave the Policy Server.

Both types of keys are kept in the Policy Server key store and distributed to Agents at runtime. By default, the key store is part of the Policy Store, but if necessary, you can create a separate key store database.

# Audit Store

The audit store is a database where the Policy Server stores audit logs containing information about authentication, authorization, and administrative events. You can use the policy store as an audit log database or configure a separate database. Configure the audit store using the Policy Server Management Console.

# Default Policy Store Objects and Schema Files

**Contents**



The CA Single Sign-On policy store is the repository for all policy–related information. All Policy Servers in a installation must share the policy store data, either directly or through replication. CA Single Sign-On is installed with tools that let administrators move policy store data from one storage facility to another.

When you install the Policy Server, you can automatically configure one of the following relational databases as a policy store:

- Microsoft SQL Server

- PostgreSQLServer

- Oracle RDBMS

Optionally, you can manually configure a policy store after installing the Policy Server. After you install the Policy Server, you can also use the Policy Server Management Console to point the Policy Server to an existing policy store.For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

In addition to policy store support, you can use a relational database to store CA Single Sign-On keys, audit logs, and session data.

## Important Considerations

Consider the following issues before configuring a policy store:

- Avoid possible policy store corruption. Be sure that the database server that is to host the policy store is configured to store objects in UTF-8 form:
  - (Oracle) Be sure that the database is configured to store objects in UTF-8 form. Oracle supports unicode within many of their character sets.
  - (SQL Server) Be sure that the database is configured using the default collation (SQL_Latin1_General_CP1_CI_AS). Using a collation that is case-sensitive can result in unexpected behaviors.
- Do not use brackets around the IP address when using IPv6 ODBC data sources or the connection fails.
  **Example:** Use fec0::9255:20c:29ff:fe47:8089 instead of [fec0::9255:20c:29ff:fe47:8089]

## Default Policy Store Objects Consideration

When you configure a policy store, the following default policy store object files are available:

- smpolicy.xml
- smpolicy-secure.xml

Both files contain the default objects that the policy store requires.

If you use the Policy Server Configuration Wizard to configure the policy store automatically, the wizard only uses smpolicy.xml.

- If you want to use smpolicy–secure.xml, configure the policy store manually.

Both files provide default security settings. These settings are available in the default Agent Configuration Object (ACO) templates that are available in the Administrative UI. The smpolicy-secure file provides more restrictive default security settings. Choosing smpolicy.xml does not limit you from using the more restrictive default security settings. You can modify the default ACO settings using the Administrative UI.

The following table summarizes the security settings for both files:

| Parameter Name | smpolicy Values | smpolicy–secure Values |
|---|---|---|
| BadCssChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadQueryChars | No value | <, >, ', ;, ), (, &, +, %00 |
| BadUrlChars | //, ./, /., /*, *., ~, \, %00-%1f, %7f-%ff, %25 | smpolicy.smdif values plus:<br><, >, ', ;, ), (, &, + |
| EnableCookieProvider | Yes | No |
| IgnoreExt | .class, .gif, .jpg, .jpeg, .png, .fcc, .scc, .sfcc, .ccc, .ntc | All smpolicy values. |
| LimitCookieProvider | No | Yes |
| ValidTargetDomain | This file does not include this parameter. | This parameter does not have a default value. Provide a valid redirection domain.<br>Example: validtargetdomain=".example.com" |

## Schema Files for Relational Databases

CA Single Sign-On provides schema files for configuring the following data stores:

- policy store
- key store
- logging database
- session store
- sample users database

**Note:** The CA Single Sign-On schema files are installed with the Policy Server. On a UNIX system, copy the schema files from *siteminder_home*/db/SQL directory to a temporary directory (C:\temp) on the system where the database resides. The *siteminder_home* place holder specifies the Policy Server installation path.

### IBM DB2 Schema Files

The following SQL Server schema files are provided in the *siteminder_home*\db\tier2\DB2 directory, where *siteminder_home* specifies the Policy Server installation path.

- **sm_db2_ps.sql**
  Creates the schema for a policy store and key store. If you are storing keys in a different database, this schema file creates the schema for the key store data.

- **sm_db2_logs.sql**
  Creates the schema for the audit logs. Be sure to edit this script (see page 346) before using it to create an audit store.

- **sm_db2_ss.sql**
  Creates the schema for the session store.

- **smsampleusers_db2.sql**
  Creates the schema for the sample users database and populates the database with sample users.

The following IBM DB2 schema file is provided in the *siteminder_home*\xps\db directory.

- **DB2.sql**
  Creates the XPS schema for a policy store.

## MySQL Schema Files

The following SQL Server schema files are provided in the *siteminder_home*\db\tier2\MySQL directory, where *siteminder_home* specifies the Policy Server installation path.

- **sm_mysql_ps.sql**
  Creates the schema for a policy store and key store. If you are storing keys in a different database, this schema file creates the schema for the key store data.

- **sm_mysql_logs.sql**
  Creates the schema for the audit logs.

- **sm_mysql_ss.sql**
  Creates the schema for a session store.

- **smsampleusers_mysql.sql**
  Creates the schema for a sample users database and populates the database with sample users.

The following MySQL schema file is in the *siteminder_home*\xps\db directory.

- **MySQL.sql**
  Creates the XPS schema for a policy store.

## PostgreSQL Server Schema Files

The following PostgreSQL schema files are provided in the *siteminder_home*\db\PostgreSQL directory, where *siteminder_home* specifies the Policy Server installation path.

- **sm_postgresql_ps.sql**
  Creates the schema for a policy store and key store. If you are storing keys in a different database, this schema file creates the schema for the key store data.

- **sm_postgresql_logs.sq**
  Creates the schema for the audit logs.

- **sm_postgresql_ss.sql**
  Creates the schema for a session store. If you do not plan on storing Unicode characters in the session store, use this file.

- **smsampleusers_postgresql.sql**
  Creates the schema for the CA Single Sign-On sample users database and populates the database with sample users.

The following PostgreSQL server schema file is provided in *siteminder_home*\xps\db:

- **PostgreSQL.sql**
  Creates the XPS schema for a policy store.

## SQL Server Schema Files

The following SQL Server schema files are provided in the *siteminder_home*\db\SQL directory, where *siteminder_home* specifies the Policy Server installation path.

- **sm_mssql_ps.sql**
  Creates the schema for a policy store and key store.
  **Note:** If you are storing keys in a different database, this schema file creates the schema for the key store data.

- **sm_mssql_logs.sql**
  Creates the schema for the audit logs.

- **sm_mssql_ss.sql**
  Creates the schema for a session store.
  **Note:** If you do not plan on storing Unicode characters in the session store, use this file.

- **sm_mssql_ss.sql.unicode**
  Creates the schema for the session store. If you plan on storing Unicode characters in the session store, use this file.

- **smsampleusers_sqlserver.sql**
  Creates the schema for the sample users database and populates the database with sample users.

The following SQL Server schema file is provided in *siteminder_home*\xps\db:

- **SQLServer.sql**
  Creates the XPS schema for a policy store.

## Oracle Schema Files

The following Oracle schema files are in the *siteminder_home*\db\SQL directory, where *siteminder_home* specifies the Policy Server installation path.

- **sm_oracle_ps.sql**
  Creates the schema for a policy store and key store. If you are storing keys in a different database, this schema file creates the schema for the key store data.

- **sm_oracle_logs.sql**
  Creates the schema for the audit logs.

- **sm_oracle_ss.sql**
  Creates the schema for a session store.

- **smsampleusers_oracle.sql**
  Creates the schema for a sample users database and populates the database with sample users.

The following Oracle schema file is provided in the *policy_server_home*\xps\db directory.

- **Oracle.sql**
  Creates the XPS schema for a policy store.

# Configure ODBC Databases as Policy Store

This section explains the following Policy Store configurations:

- Configure a MySQL Policy Store (see page 218)
- Configure a PostgreSQL Server Policy Store (see page 230)
- Configure a SQL Server Policy Store (see page 242)
- Configure an Oracle Policy Store (see page 254)
- Configure an IBM DB2 Policy Store (see page 274)

## Configure a MySQL Policy Store

**Contents**



A MySQL policy store can also function as:

- A key store
- An audit logging database

> **Note:** Session information is stored in a separate database. Do not use the policy store to store session information.

Using a single database simplifies administration tasks. The following sections provide instruction on how to configure a single database server.

## Before You Begin

1. Verify that MySQL uses the Latin1 or UTF8 character set. Use the UTF8 character set to support Unicode characters.
2. Confirm that the MySQL database acting as the policy store is accessible from the Policy Server host system.
3. Create the database instance for the data store, using the vendor–specific user interface.
   - To create a database instance for Unicode characters, use the character set and collation for UTF8.
   - To create a database instance for non-Unicode characters, use the character set and collation for Latin1.

## Gather Database Information

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store:

- **Database host**—Identify the name of the database host system.
- **Database name**—Identify the name of the database instance that is to function as the policy store or data store.
- **Database port**—Identify the port on which the database is listening.
- **Administrator account**—Identify the login ID of an administrator account with permission to manage objects in the database.
- **Administrator password** —Identify the password for the administrator account.

## Create the CA Single Sign-On Schema

You create the CA Single Sign-On schema so that the MySQL database can store the policy, key, and audit logging information.

**Follow these steps:**

1. Start the Query Browser and log in as the person who administers the Policy Server database.
2. Select the database instance from the database list.
3. Navigate to the following location:
   *siteminder_home*\db\tier2\MySQL.
   - **siteminder_home**
     Specifies the Policy Server installation path.
4. Open *one* of the following files in a text editor:

- To store Unicode characters in the policy store, open sm_mysql_ps.sql.unicode.
- To store non-Unicode characters in the policy store, open sm_mysql_ps.sql.

5. Locate the following lines in the sm_mysql_ps file:

```
DROP FUNCTION IF EXISTS `databaseName`.`getdate` $$
CREATE FUNCTION `databaseName`.`getdate` () RETURNS DATE
```

6. Replace each instance of 'databaseName' with the name of the database functioning as the policy store.
7. After you replace the databaseName instances, copy the contents of the entire file.
8. Paste the file contents into a query and execute the query.
   The policy and key store schema are added to the database.
9. Navigate to the following location:
   *siteminder_home*\xps\db\Tier2DirSupport\MySQL
10. Open *one* of the following files in a text editor and copy the contents of the entire file:
    - To store Unicode characters in the policy store, open MySQL.sql.unicode.
    - To store non-Unicode characters in the policy store, open MySQL.sql.
11. Paste the schema from the appropriate MySQL file into a query and execute the query.
    The policy store schema is extended.
12. To use the policy store as an audit logging database, repeat steps three and four but use the following logging schema file:
    sm_mysql_logs.sql
    The database can store CA Single Sign-On data.
    **Note:** You are not required to configure the policy store to store more CA Single Sign-On data. You can configure individual databases to function as a separate audit log database, key store, and session store.

## Configure a MySQL Data Source for CA Single Sign-On

You configure a data source to let the Policy Server communicate with the CA Single Sign-On data store.

**Note**: If you are using MySQL 5.1.x, ensure that you assign the TRIGGER permission to the user name that is used to create the DSN.

## Create a MySQL Data Source on Windows

You create a MySQL data source for the MySQL wire protocol driver.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Do one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
3. Click System DSN.
4. Click Add.
5. Scroll down and select CA Single Sign-On MySQL Wire Protocol and click Finish.
6. Complete the following steps in the General tab:
   a. Enter a data source name in the Data Source Name field.
   **Example:**

   ```
   CA SiteMinder® MySQL Wire Data Source
   ```

   b. Enter the name of the MySQL database host system in the Host Name field.
   c. Enter the port on which the MySQL database is listening in the Port Number field.
   d. Enter the name of the MySQL database in the Database Name field.
7. Click Test Connect.
8. Click OK.
   The data source is created and appears in the System Data Sources list.

**Note:** You can now point the Policy Server to the CA Single Sign-On data store.

## Create a MySQL Data Source on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming mysqlwire.ini to system_odbc.ini. The mysqlwire.ini file is located in *siteminder_home*/db.

- **siteminder_home**
  Specifies the Policy Server installation path.

This system_odbc.ini file contains all of the names of the available ODBC data sources and the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add other data sources to this file, such as defining other ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** The value of the first line of data source entry is required when you configure the database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver that is loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

Adding a MySQL Server Data source includes:

- A new data source name in the [ODBC Data Sources] section of the file.
- A section that describes the data source using the same name as the data source.

Update the system_odbc.ini file when creating a new service name. You have entries for the MySQL driver under [CA Single Sign-On Data Source].

Again, to configure a MySQL Server data source, you create the system_odbc.ini file by renaming mysqlwire.ini to system_odbc.ini.

## Create the MySQL Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.

- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Enter the following line under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 MySQL Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSmysql27.so
   Description=DataDirect 7.1 MySQL Wire Protocol
   Database=database_nameHostName=host_nameLogonID=root_userPassword=root_user_pass
   wordPortNumber=mysql_port
   ```

   - *nete_ps_root*
     Specifies the Policy Server installation path. Enter this value as an explicit path, rather than one with an environment variable.
     **Example:** /export/smuser/siteminder

   - *database_name*
     Specifies the name of the MySQL database that is to function as the data store.

   - *host_name*
     Specifies the name of the MySQL database host system.

   - *root_user*
     Specifies the login ID of the MySQL root user.

   - *root_user_password*
     Specifies the password for the MySQL root user.

- *mysql_port*
  Specifies the port on which the MySQL database is listening.

4. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can access the CA Single Sign-On data in the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console and click the Data tab.
2. Select the following value from the Storage list:

   `ODBC`
3. Select the following value from the Database list:

   `Policy Store`
4. Enter the name of the data source in the Data Source Information field.
   - (Windows) The entry must match the name that you entered in the Data Source Name field when you created the data source.
   - (UNIX) The entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure to enter the correct value.
5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
6. Specify the maximum number of database connections that are allocated to CA Single Sign-On.

   **Note:** We recommend retaining the 25 connection default for best performance.
7. Click Apply to save the settings.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `ODBC`
10. Select the following option:

    `Use the Policy Store database`

11. Select the following value from the Database list:

    `Audit Logs`

12. Select the following value from the Storage list:

    `ODBC`

13. Select the following option:

    `Use the Policy Store database`

14. Click Apply to save the settings.

15. Click Test Connection to verify that the Policy Server can access the policy store.

16. Click OK.
    The Policy Server is configured to use the database as a policy store, key store, and logging database.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   `smreg -su password`

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

- The password must contain at least six (6) characters and cannot exceed 24 characters.
- The password cannot include an ampersand (&) or an asterisk (*).
- If the password contains a space, enclose the passphrase with quotation marks.

**Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.

- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure a PostgreSQL Server Policy Store

**Contents**



Prerequisites:

- Be sure that the PostgreSQL database instance that is to contain the data is accessible from the Policy Server system.

- Create the database instance for the data store.
  **Example:**

  ```
  smdatastore
  ```

**Follow these steps:**

1. Gather the PostgreSQL database information.
2. Create the schema.
3. Configure a PostgreSQL data source for your operating platform:
   **Windows:** Create a PostgrSQL data source.
   **UNIX:**
   - Create a PostgreSQL data source on UNIX systems.
   - Configure the PostgreSQL wire protocol driver.
4. Point the Policy Server to the database.
5. Set the super user password.
6. Import the policy store data definitions.
7. Import the default policy objects.
8. Restart the Policy Server.
9. Prepare for the Administrative UI registration.

## Gather PostgreSQL Database Information

Configuring a single PostgreSQL server database as a policy store or any other type of data store requires specific database information.

Before configuring the policy or data store, gather the following information:

- **Database instance name**
  Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account name and password**
  Determine the user name and password of an account with privileges to create, read, modify, and delete objects in the database.
- **Data source name**
  Determine the name you will use to identify the data source.
  **Example:** SM PostgreSQL Server Wire DS.
- **PostgreSQL server name**
  Determine the name of the PostgreSQL server database that contains the instance that is to function as the policy store.
- **Policy Server root**
  Determine the explicit path to where the Policy Server is installed.
- **IP Address**
  Determine the IP Address of the PostgreSQL server database.

## Create the CA Single Sign-On Schema for PostgreSQL

You create the schema so that SQL Server database can store policy, key, session, and audit logging information.

**Follow these steps:**

1. Start the PostgreSQL client and log in as the person who administers the Policy Server database.
2. Select the database instance from the database list.
3. Open *policy_server_home*/db/SQL/sm_postgresql_ps.sql in a text editor and copy the contents of the entire file.
4. Paste the schema from sm_postgresql_ps.sql into the query and execute the query.
   The policy and key store schema is added to the database.
5. Open *policy_server_home*/xps/db/PostgreSQL.sql in a text editor and copy the contents of the entire file.
6. Paste the schema from PostgreSQL.sql into the query, and execute the query.
   The policy store schema is extended.
7. Repeat steps three and four to use the policy store as an audit logging database. Use the following schema file:
   sm_postgresql_logs.sql

   **Note:** You are not required to configure the policy store to store additional CA Single Sign-On data. You can configure individual databases to function as a separate audit log database, key store, and session store.

   The database can store data.

## Create a PostgreSQL Data Source on Windows

ODBC requires that you configure a data source for the SQL Server wire protocol.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32 - bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64 - bit Windows operating system:
     a. Navigate to C:\Windows\SysWOW64.
     b. Double - click odbcad32.exe.

The ODBC Data Source Administrator appears.

2. Click the System DSN tab.

3. Click Add.

4. Select CA Single Sign-On PostgreSQL Server Wire Protocol and click Finish.

5. Enter the data source name in the Data Source Name field.
   **Example:** CA Single Sign-On Data Source.
   **Note:** Take note of your data source name. This information is required as you configure your database as a policy store.

6. Enter the name of the PostgreSQL host system in the Server field.

7. Enter the database name in the Database Name field.

8. Click Test.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The PostgreSQL data source is configured and appears in the System Data Sources list.

## Create a PostgreSQL Data Source on UNIX

You configure the CA Single Sign-On ODBC data sources using a system_odbc.ini file, which you create by renaming postgresqlwire.ini, located in *policy_server_installation*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of data source entry, which is [CA Single Sign-On Data Source], take note of the change because you use this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by the policy server. The remaining attributes are specific to the driver.

Adding a PostgreSQL server data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You have entries for the Oracle or SQL drivers under [CA Single Sign-On Data Source].

## Configure the PostgresSQL Server Wire Protocol Driver

You configure the wire protocol driver to specify the settings CA Single Sign-On uses to connect to the database.

Note: This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it system_odbc.ini. The file you rename depends on the database vendor you are configuring as a data store.

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in siteminder_home/db

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[CA Single Sign-on Data Source]**
  Specifies the settings to connect to the database functioning as the policy store.
- **[CA Single Sign-on Logs Data Source]**
  Specifies the settings to connect to the database functioning as the audit log database.
- **[CA Single Sign-on Keys Data Source]**
  Specifies the settings to connect to the database functioning as the key store.
- **[CA Single Sign-on Session Data Source]**
  Specifies the settings to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 PostgrSQL Server Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information:

```
Driver=nete_ps_root/odbc/lib/NSpsql27.so
Description=DataDirect 7.1 PostgreSQL Server Wire Protocol
Database=SiteMinder Data
Address=myhost, 1433
QuotedId=No
AnsiNPW=No
```

**Note:** When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

- ***nete_ps_root***
  Specifies the explicit path of the Policy Server installation, rather than a path with an environment variable.
  **Example:** export/smuser/siteminder

- **CA Single Sign-On Data**
  Specifies the PostgreSQL Server database instance name.

- **myhost**
  Specifies the IP Address of the PostgreSQL Server database.

- **1433**
  Represents the default listening port for PostgreSQL Server.

4. Edit the [ODBC] section as follows:

   ```
   TraceFile=nete_ps_root/db/odbctrace.out
   TraceDll=nete_ps_root/odbc/lib/NStrc27.so
   InstallDir=nete_ps_root/odbc
   ```

   - *nete_ps_root*
     Specifies the explicit path to the Policy Server installation directory. This path cannot contain an environment variable.

5. Save the file.

The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can access the data in the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console and click the Data tab.
2. Select the following value from the Storage list:

   ODBC

3. Select the following value from the Database list:

```
Policy Store
```

4. Enter the name of the data source in the Data Source Information field.
   - (Windows) The entry must match the name that you entered in the Data Source Name field when you created the data source.
   - (UNIX) The entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure to enter the correct value.
5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
6. Specify the maximum number of database connections that are allocated to CA Single Sign-On.

   **Note:** We recommend retaining the 25 connection default for best performance.

7. Click Apply to save the settings.
8. Select the following value from the Database list:

   ```
   Key Store
   ```

9. Select the following value from the Storage list:

   ```
   ODBC
   ```

10. Select the following option:

    ```
    Use the Policy Store database
    ```

11. Select the following value from the Database list:

    ```
    Audit Logs
    ```

12. Select the following value from the Storage list:

    ```
    ODBC
    ```

13. Select the following option:

    ```
    Use the Policy Store database
    ```

14. Click Apply to save the settings.
15. (Optional) Click Test Connection to verify that the Policy Server can access the policy store.
16. Click OK.
    The Policy Server is configured to use the database as a policy store, key store, and logging database.

## Set the CA Single Sign-On Super User Password

The default administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default superuser for day- to-day operations. Use the default super user to:

- Access the Administrative UI for the first - time.
- Manage CA Single Sign-On utilities for the first - time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case - sensitive. The password is not case - sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - *siteminder_home*
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - *siteminder_home*
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy - secure.xml, run the following command:

     ```
     XPSImport smpolicy - secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy - secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration.

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing smpolicy.xml makes available federation and Web Service Variables functionality.

## Enable the Advanced Authentication Server

Enable the advanced authentication server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Leave all the check boxes in the first screen of the wizard *cleared*.
3. Click Next.
4. Create the master encryption key for the advanced authentication server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

5. Complete the rest of the Policy Server configuration wizard.
   The advanced authentication server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands respectively to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default CA Single Sign-On super user account (siteminder) to log into the Administrative UI for the first - time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
   c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   - *passphrase*
     Specifies the password for the default super user account (siteminder).

     **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

   - **-adminui - setup**
     Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   - **-t *timeout***
     (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server

denies the registration request when the timeout value is exceeded.
**Unit of measurement:** minutes
**Default:** 240 (4 hours)
**Minimum:** 15
**Maximum:** 1440 (24 hours)

- **-r *retries***
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c *comment***
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l *log path***
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e *error_path***
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure a SQL Server Policy Store

**Contents**



A single SQL Server database can function as a:

- policy store
- key store
- logging database

**Note:** Store session information in a separate database. Do not use the policy store to store session information.

Using a single database simplifies administration tasks. The following sections provide instruction on how to configure a single database server to store data.

Consider the following items:

- You can configure a SQL Server policy store manually or you can use the Policy Server installer to configure the policy store automatically.
- The database must be installed on a Windows system.Additionally, the CA Single Sign-On schema files are installed with the Policy Server. If the Policy Server is installed on a UNIX system, copy the schema files from the *policy_server_home*/db/SQL directory to a temporary directory on the Windows system.

## Create the Database Instance

Using the SQL Server Enterprise Manager, create the database instance for the CA Single Sign-On data store.

**Example:**

```
smdatastore
```

## Gather Database Information

Configuring a single SQL Server database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

**Note:** Information prefixed with (W) indicates that the information is only required if the Policy Server is installed on a Windows system; (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. Different information is required when configuring the SQL Server data source.

- **Database instance name**
  Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account name and password**
  Determine the user name and password of an account with privileges to create, read, modify, and delete objects in the database.
- **(W) Data source name**
  Determine the name you will use to identify the data source.
  **Example:** SM SQL Server Wire DS.
- **(W) SQL Server name**
  Determine the name of the SQL Server database that contains the instance that is to function as the policy store.
- **(U) Policy Server root**
  Determine the explicit path to where the Policy Server is installed.
- **(U) IP Address**
  Determine the IP Address of the SQL Server database.

## Create the CA Single Sign-On Schema

You create the CA Single Sign-On schema so that SQL Server database can store policy, key, and audit logging information.

The following warnings are displayed when running the policy store and audit logging schema files. The warnings do not affect the policy store configuration:

- Warning: The table 'smvariable5' has been created but its maximum row size (8746) exceeds the maximum number of bytes per row (8060). INSERT or UPDATE of a row in this table will fail if the resulting row length exceeds 8060 bytes.
- Warning: The table 'smodbcquery4' has been created but its maximum row size (64635) exceeds the maximum number of bytes per row (8060). INSERT or UPDATE of a row in this table will fail if the resulting row length exceeds 8060 bytes.
- Warning: The table 'smaccesslog4' has been created but its maximum row size (9668) exceeds the maximum number of bytes per row (8060). INSERT or UPDATE of a row in this table will fail if the resulting row length exceeds 8060 bytes.

**Follow these steps:**

1. Start the Query Analyzer and log in as the person who administers the Policy Server database.
2. Select the database instance from the database list.
3. Open sm_mssql_ps.sql in a text editor and copy the contents of the entire file.
   **Default Location**: *installation_path*/db/SQL
4. Paste the schema from sm_mssql_ps.sql into the query and execute the query.
   The policy and key store schema is added to the database.
5. Open SQLServer.sql in a text editor and copy the contents of the entire file.
   **Default Location**: *installation_path*/xps/db
6. Paste the schema from SQLServer.sql into the query, and execute the query.
   The policy store schema is extended.
7. Repeat steps three and four to use the policy store as an audit logging database. Use the following schema file:
   sm_mssql_logs.sql

   **Note:** You are not required to configure the policy store to store additional CA Single Sign-On data. You can configure individual databases to function as a separate audit log database, key store, and session store.

   The database can store CA Single Sign-On data.

## Configure a SQL Server Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## SQL Server Authentication Mode Considerations

CA Single Sign-On data sources do not support Windows authentication. Configure the CA Single Sign-On data source with the credentials of a user that is stored in the database.

**Note:** For more information about SQL Server authentication modes, see the vendor–specific documentation.

## Create a SQL Server Data Source on Windows

ODBC requires that you configure a data source for the SQL Server wire protocol.

**Note:** This procedure only applies if the Policy Server is installed on a Windows System.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab.
   System data source settings appear.
3. Click Add.
   The Create New Data Source dialog appears.
4. Select CA Single Sign-On SQL Server Wire Protocol and click Finish.
   The ODBC SQL Server Wire Protocol Driver Setup dialog appears.
5. Enter the data source name in the Data Source Name field.
   **Example:** CA Single Sign-On Data Source.
   **Note:** Take note of your data source name. This information is required as you configure your database as a policy store.
6. Enter the name of the SQL Server host system in the Server field.

7. Enter the database name in the Database Name field.

8. Click Test.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The SQL Server data source is configured and appears in the System Data Sources list.

## Create a SQL Server Data Sources on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of data source entry, which is [CA Single Sign-On Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-On. The remaining attributes are specific to the driver.

Adding a MS SQL Server Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the Oracle or SQL drivers under [CA Single Sign-On Data Source].

Again, to configure a MS SQL Server data source, you must first create a system_odbc.ini file in the *policy_server_installation*/db directory. To do this, you need to rename sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini.

## Configure the SQL Server Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini

- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 SQL Server Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSsqls27.so
   Description=DataDirect 7.1 SQL Server Wire Protocol
   Database=SiteMinder DataAddress=host_ip, 1433
   QuotedId=No
   AnsiNPW=No
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation, rather than a path with an environment variable.
     **Example:** export/smuser/siteminder

- *SiteMinder Data*
  Specifies the SQL Server database instance name.

- *host_ip*
  Specifies the IP Address of the SQL Server database.

- **1433**
  Represents the default listening port for SQL Server.

4. If you are using Microsoft SQL Server 2008 to function as any CA Single Sign-On store, edit the [ODBC] section as follows:

   ```
   TraceFile=nete_ps_root/db/odbctrace.out
   TraceDll=nete_ps_root/odbc/lib/NStrc27.so
   InstallDir=nete_ps_root/odbc
   ```

   - *nete_ps_root*
     Specifies the explicit path to the Policy Server installation directory. This path cannot contain an environment variable.

5. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can access the CA Single Sign-On data in the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console and click the Data tab.

2. Select the following value from the Storage list:

   ```
   ODBC
   ```

3. Select the following value from the Database list:

   ```
   Policy Store
   ```

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) The entry must match the name that you entered in the Data Source Name field when you created the data source.

   - (UNIX) The entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure to enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

6. Specify the maximum number of database connections that are allocated to CA Single Sign-On.

**Note:** We recommend retaining the 25 connection default for best performance.

7. Click Apply to save the settings.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `ODBC`
10. Select the following option:

    `Use the Policy Store database`
11. Select the following value from the Database list:

    `Audit Logs`
12. Select the following value from the Storage list:

    `ODBC`
13. Select the following option:

    `Use the Policy Store database`
14. Click Apply to save the settings.
15. Click Test Connection to verify that the Policy Server can access the policy store.
16. Click OK.
    The Policy Server is configured to use the database as a policy store, key store, and logging database.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.

- *siteminder_home*
  Specifies the Policy Server installation path.

**Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

**Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

   - ***siteminder_home***
     Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```
   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```
     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).
   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```
   - To import federation functionality, run the following command:

     ```
     XPSImport fedpolicy-12.5.xml -npass
     ```

   The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

> **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  > **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Configure an Oracle Policy Store

**Contents**

A single Oracle database can function as a:

- policy store
- key store
- logging database

Using a single database simplifies administrative tasks. The following sections provide instruction on how to configure a single database server to store CA Single Sign-On data.

You can configure an Oracle policy store manually or use the Policy Server installer to configure the policy store automatically.

To configure a single Oracle database as a policy store, key store, and logging database, complete the following procedures:

## Prerequisites for an Oracle 10g Database

After installing the Oracle 10g database, complete the following prerequisites:

- Create a table space for the policy store.
- Create a user with appropriate privileges to manage this table space in the database.

## Create an Oracle 10g Table Space for the Policy Store

Creating a table space for the policy store is a prerequisite for an Oracle 10g database only.

**Follow these steps:**

1. In the Oracle Enterprise Manager 10g Database Control, log in as the SYSDBA user with appropriate privileges to manage the Oracle database.
2. On the Oracle global database's configuration screen, select Administration, Tablespaces.
3. On the Tablespaces screen, click Create.
4. On the Create Tablespaces screen, enter a table space name, and click ADD.
   **Example:** NETE_TB
5. On the Create Tablespaces: Add Datafile screen:

a. Enter a file name.
   Example: NETE_TB

b. Specify the file size.
   Example: 100 MB

c. Click Continue.

Oracle creates the table space and displays it on the Tablespaces screen.

Complete the prerequisites by creating a user to manage the table space for the policy store.

## Create an Oracle 10g User to Manage the Policy Store's Table Space

Creating a user to manage table space for the policy store is a prerequisite for an Oracle 10g database only.

**Follow these steps:**

1. On the Oracle global database's configuration screen, select Administration, Users.
2. On the Create Tablespaces screen, click Create.
3. On the Create User screen, enter the:
   - Name for the user.
     Example: NETE
   - Password for the user.
   - Default Tablespace that you created.
   - Temporary tablespace.
     **Example:** TEMP
4. Click Roles.
5. Select Modify.
6. On the Modify Roles screen:
   a. Select CONNECT and RESOURCE as a roles for this user.
   b. Click Apply.
7. Start sqlplus in a command window, by entering:
   a. sqlplus
   b. the credentials for the policy store user created on the Create User screen.

You have completed the prerequisites for an Oracle 10g database, and can now configure a CA Single Sign-On data store for the database.

## Prerequisites for an Oracle 12c Database

After you install the Oracle 12c Database, perform the following steps:

1. Create a table space for the policy store.
2. Create a user with the following privileges to manage the table space in the database:

- Connect
- Resource
- Unlimited Tablespace

## Gather Database Information

Configuring a single Oracle database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Information prefixed with (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. This information is required when configuring Oracle data source for UNIX.

## Required Information

Gather the following required information before configuring a supported Oracle or Oracle RAC database as a policy store or any other type of CA Single Sign-On data store:

- (U) **Policy Server installation path**—Identify the explicit path to where the Policy Server is installed.
- **Data source**—Determine the name you will use to identify the Oracle data source.
  **Example:** SM Oracle Server Wire DS.
- **Database administrative account**—Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.

  **Note:** Ensure the administrative account does not have the DB role. Audit-based reports will not return correct results if the administrative account has the DB role.

- **Database administrative Password**—Determine the password for the Administrative account.

## Oracle Database Information

Gather the following information only if you are configuring a supported Oracle database as a policy store or any other type of CA Single Sign-On data store:

- **Oracle machine name**—Determine the name of the machine on which the Oracle database is installed.

- **Oracle instance service name**—Determine the service name of the database instance to which you will connect. The tnsnames.ora file specifies service names.

- **Oracle port number**—Determine the port number on which the Oracle database is listening.

## Oracle RAC Database (without SCAN) Information

Gather the following information if you are configuring a supported Oracle RAC database (without SCAN functionality configured) as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name—**Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
(Description =
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername1)(PORT=1521
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername2)(PORT=1521)
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername3)(PORT=1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA=
(SERVER = DEDICATED)
(SERVER_NAME = SMDB))
)
```

- **Oracle RAC node service names**—Determine the service names for each node in the system.

- **Oracle RAC node IP addresses**—Determine the IP Address of each node in the Oracle RAC system.

  **Note:** If you are using Oracle RAC 10g, determine the virtual IP address of each node in the system.

- **Oracle RAC node port numbers**—Determine the port number for each node in the Oracle RAC system.

## Oracle RAC Database (Using SCAN) Information

The Oracle RAC Single Client Access Name (SCAN) feature provides a single name for clients to access any Oracle Database running in a cluster.

Gather the following information if you are configuring an Oracle RAC database with SCAN functionality as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name**—Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SMDB)
    )
  )
```

- **Oracle RAC SCAN Address**—Determine the FQDN of the Oracle RAC system SCAN.

- **Oracle RAC SCAN port number**—Determine the port number for the Oracle RAC system SCAN.

## Create the CA Single Sign-On Schema

You create the schema so a single Oracle database can store policy, key, and audit logging information.

**Follow these steps:**

1. Log in to Oracle with sqlplus or some other Oracle utility as the user who administers the Policy Server database information.

   **Note:** We recommend that you do not create the schema with the SYS or SYSTEM users. If necessary, create an Oracle user, such as SMOWNER, and create the schema with that user.

2. Import the following script:

   ```
   $NETE_PS_ROOT/db/sql/sm_oracle_ps.sql
   ```

   **Note:** Environment variables may not function in the SQL utility of Oracle. If you experience problems importing the script using the utility, specify an explicit path.

   The policy store and key store schema is added to the database.

3. Import the following script

   ```
   $NETE_PS_ROOT/xps/db/Oracle.sql
   ```

   The policy store schema is extended.

4. Import the following script to use the policy store as an audit logging database:

   ```
   sm_oracle_logs.sql.
   ```

   **Note:** You are not required to configure the policy store to store additional CA Single Sign-On data. You can configure individual databases to function as a separate audit log database, key store, and session store.
   The database can store CA Single Sign-On data.

## Configure an Oracle Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source for the Oracle wire protocol driver.

## Create an Oracle Data Source on Windows

Create an ODBC data source for an Oracle database.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the *install_home*\Windows\SysWOW64.

     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the machine name where the Oracle database is installed in the Host Name field.

6. Enter the port number where the Oracle database is listening on the machine in the Port Number field.

7. Enter the name of the Oracle instance to which you want to connect in the SID field.

   **Note:** The service name is specified in the tnsnames.ora file. The SID is the system identifier for the database instance. The tnsnames.ora file contains service names and details that Oracle uses to identify and connect to Oracle instances.

   **Example:** if the tnsnames.ora file contains the following entry for an Oracle instance, you enter instance1 in the SID field:

```
instance1 =
    (Description=
    (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
    (Connect_DATA_ = (SID = SIDofinstance1))
    )
```

8. Click Test Connection.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The Oracle data source is configured for the wire protocol driver.

## Create an Oracle RAC (no SCAN) Data Source on Windows

Create an ODBC data source for an Oracle RAC database that does not use the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.

   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the C:\Windows\SysWOW64.

     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the IP Address of the first node in the Oracle RAC system in the Host field.
   Oracle RAC 10g: Enter the virtual IP Address.

6. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains 3 nodes:

```
SMDB=
    (Description =
(ADDRESS = (Protocol = TCP)(HOST = nete_servername1)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername2)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername3)(PORT = 1521))
(LOAD_BALANCE = yes)
```

```
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = SMDB)
)
```

7. Click the Failover tab.
   Failover settings appear.

8. Specify the host name or virtual IP Address, port number, and service name for the remaining Oracle RAC nodes in the environment in the Alternate Servers field.

   **Note:** The ServiceName is the service name for the entire Oracle RAC system.

9. Specify the AlternateServers to provide connection failover to the other Oracle nodes if the primary server is not accepting connections. The entry should have the following format:
   (HostName=nete_servername2:PortNumber=1521:ServiceName=nete_servicename[,...])

10. Select LoadBalancing.

11. Click OK
    The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle RAC SCAN Data Source on Windows

Create an ODBC data source for an Oracle RAC database that uses the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA CA Single Sign-on Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

**Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the FQDN or IP Address of the SCAN in the Host field.

6. Enter the port number of the SCAN in the Port Number field.

7. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains the SCAN:

```
SMDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.rac.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORCL)
    )
  )
```

8. Click OK
   The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle Data Source on UNIX Systems

You configure the names of available ODBC data sources and the attributes that are associated with these data sources in the system_odbc.ini file.

**To create the system_odbc.ini file:**

1. Navigate to *policy_server_installation*/db

2. Rename oraclewire.ini to "system_odbc.ini".

Customize the system_odbc.ini file for each site. You can also add more data sources to this file, such as defining extra ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the “=” refers to a subsequent section of the file describing each individual data source. After the “=” is a comment field.

**Note:** If you modify of the first line of the data source entry ([CA Single Sign-On Data Source]), take note of the change. This value is required to configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

**To add an Oracle Data source**:

1. Define a new data source name in the [ODBC Data Sources] section of the file.

2. Add a section that describes the data source using the same name as the data source.

To create a service name or use a different driver, edit the system_odbc.ini file. Entries for the SQL Server or Oracle drivers belong under [CA Single Sign-On Data Source].

## Configure the Oracle Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

```
Driver=nete_ps_root/odbc/lib/NSora27.so (http://nsora27.so/)Description=DataDirect
 7.1 Oracle Wire Protocol
LoginID=uidPassword=pwdHostName=host_namePortNumber=1521
SID=server_idCatalogOptions=0
ProcedureResults=0
EnableDisableParam=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation.
   - *uid*
     Specifies the user name of the database account that has full access rights to the database.
   - *pwd*
     Specifies the password for the database account that has full access rights to the database.
   - *host_name*
     Specifies the name of the Oracle database host system.
     *server_id*
     Specifies the Oracle instance service name (SID). The SID is the system identifier for the database instance.

   **Example:** In the following sample tnsnames.ora file, the value instance1 is the SID

```
instance1 =
(Description =
(ADDRESS = (Protocol = TCP)(Host = myhost)(Port = 1521)
(CONNECT_DATA = (SID = instance1))
)
```

3. Save the file.

The Oracle wire protocol driver is configured.

## Configure the Oracle Wire Protocol Driver for Oracle RAC without SCAN

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini

- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.

- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   - Add ServiceName=*nete_servicename*
   - Add AlternateServers=
   - Add Loadbalancing=1
   - Remove or comment SID=nete_serverid

   The modified text for the data source should appear as follows:

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so
   Description=DataDirect 7.1 Oracle Wire Protocol
   Logon=uidPassword=pwdHostName=server_name1PortNumber=1521
   ServiceName=service_nameCatalogOptions=0
   ProcedureRetResults=0
   EnableDescribeParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   AlternateServers=
   LoadBalancing=1
   ```

- *nete_ps_root*
  Specifies an explicit path to the directory where Policy Server is installed.

- *uid*
  Specifies the user name of the database account that has full access rights to the database.

- *pwd*
  Specifies the password for the database account that has full access rights to the database.

- *server_name1*
  Specifies the IP Address of the first Oracle RAC node.
  (Oracle 10g) Specifies the virtual IP Address of the first Oracle RAC node.

- *service_name*
  Specifies the Oracle RAC system service name for the entire RAC system.

- **AlternateServers=**
  If the primary server is not accepting connections, specifies the connection failover to the other Oracle nodes.
  **Example:** (HostName=nete_servername2:PortNumber=1521:
  ServiceName=nete_servicename[,...])

- **LoadBalancing=1**
  Turns on client load balancing, which helps to distribute new connections to keep RAC nodes from being overwhelmed with connection requests. When enabled, the order in which primary and alternate database servers are accessed is random.

3. Save the file.
   The Oracle wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can access the CA Single Sign-On data in the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console and click the Data tab.

2. Select the following value from the Storage list:

   ```
   ODBC
   ```

3. Select the following value from the Database list:

   ```
   Policy Store
   ```

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) The entry must match the name that you entered in the Data Source Name field when you created the data source.

- (UNIX) The entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure to enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
6. Specify the maximum number of database connections that are allocated to CA Single Sign-On.

   **Note:** We recommend retaining the 25 connection default for best performance.

7. Click Apply to save the settings.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `ODBC`
10. Select the following option:

    `Use the Policy Store database`
11. Select the following value from the Database list:

    `Audit Logs`
12. Select the following value from the Storage list:

    `ODBC`
13. Select the following option:

    `Use the Policy Store database`
14. Click Apply to save the settings.
15. Click Test Connection to verify that the Policy Server can access the policy store.
16. Click OK.
    The Policy Server is configured to use the database as a policy store, key store, and logging database.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.

- ***siteminder_home***
  Specifies the Policy Server installation path.

2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.

  - **siteminder_home**
    Specifies the Policy Server installation path.

- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.

2. Import one of the following files:

   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

     Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

   - To import Option Pack functionality, run the following command:

     ```
     XPSImport ampolicy.xml -npass
     ```

   - To import federation functionality, run the following command:

```
XPSImport fedpolicy-12.5.xml -npass
```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.
3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.
4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

**Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure an IBM DB2 Policy Store

**Contents**



A single IBM DB2 database can function as a:

- policy store
- key store
- logging database

**Note:** Store session information in a separate database. Do not use the policy store to store session information.

Using a single database simplifies administrative tasks. The following sections provide instruction on how to configure a single database server to store CA Single Sign-On data.

Complete the following procedures to configure a single IBM DB2 database as a policy store, key store, and logging database.

Be sure that you have gathered the required database information before beginning. Some of the following procedures require this information.

## Prerequisite Database Settings

Configure the following database instance settings:

- If you are configuring only a policy store, set the table space page size (page_size) and the buffer pool page size settings to at least 16k. The default DB2 value for each setting is not sufficient for the policy store schema.

  To create an instance with 16k page size, run the following command from the DB2 admin command prompt:

  ```
  db2 create database <database> using codeset UTF-8 territory en PAGESIZE 16384
  ```

- If you are configuring a policy store and an audit store, set the table space page size (page_size) and the buffer pool page size settings to at least 32k. The default DB2 value for each setting is not sufficient for the policy and audit store schemas.

## Gather Database Information

Configuring a single IBM DB2 database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Consider the following items:

- Information that is prefixed with a W represents a Windows requirement.
- Information that is prefixed with a U represents a UNIX requirement.

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store. You can use the IBM DB2 Information Worksheet to record your values.

- **Database instance name** —Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account** —Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.
- **Administrative password** —Determine the password for the Administrative account.
- **IP address**—Determine the IP address of the database host system.
- **Tcp port**—Determine the port on which the database is listening.
- (W) **Data source name** —Determine the name that is to identify the data source.
- (U) **Policy Server root** —Determine the explicit path to where the Policy Server is installed.
- (U) **Package** —Determine the name of the package that is to process dynamic SQL.
- (U) **Package owner**—Determine the AuthID assigned to the package. The AuthID must have the authority to execute all SQLs in the package.

- (U) **Grant AuthID**—If you want to restrict execute privileges for the package, determine the AuthID that is granted execute permissions for the package.
  **Default wire protocol setting:** Public

- (U) **Isolation level**—Determine the method by which the system acquires and releases locks.
  **Default wire protocol setting:** CURSOR_STABILTY

- (U) **Dynamic sections**—Determine the number of sections that the wire protocol driver package can prepare for a single user.
  **Default wire protocol setting:** 100

## Create the Schema

**Follow these steps:**

1. Navigate to *siteminder_home*\db\tier2\DB2.
   *siteminder_home* specifies the Policy Server installation path.

2. Open the **sm_db2_ps.sql** file in a text editor and copy the contents of the entire file:
   This file specifies the schema for a policy or key store in a DB2 database.

3. Paste the file contents into a query and execute the query.
   The policy and key store schema is created in the DB2 database.

4. (Optional) Repeat steps two and three to create the audit log or sample users schema in the DB2 database:

   - **sm_db2_logs.sql**
     Specifies the schema for an audit log store in a DB2 database. Edit this script (see page 346 ) before creating an audit store.

   - **smsampleusers_db2.sql**
     Specifies the schema for sample users in a DB2 database and populates the database with the sample users.

   The corresponding schema is created in the DB2 database.

   **Note:** Using the policy store to store key, audit, and sample users is optional. You can use separate databases to function as these types of data stores individually.

5. Copy the DB2.sql schema file from one of the following locations to the DB2 host system.
   **Location:**
   In case of a new installation, you can locate the schema file in the following directory:
   *siteminder_home*\xps\db\Tier2DirSupport.
   In case of an upgrade scenario, the schema file is located in the following directory:
   *siteminder_home*\xps\db\schema_extension\db\IBM DB2.
   siteminder_home specifies the Policy Server installation path.

6. Open a command prompt and run the following command:

```
db2 -td@ [-v] -f path\DB2.sql
```

*path* specifies the path to the DB2 schema file.

7. The policy store schema is created.

## Configure an IBM DB2 Data Source for CA Single Sign-On

If you are using ODBC, configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## Create a DB2 Data Source on Windows Systems

When using ODBC, you can create a DB2 data source for the DB2 wire protocol driver.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab and click Add.
3. Scroll down and select CA Single Sign-On DB2 Wire Protocol and click Finish.
4. In the ODBC DB2 Wire Protocol Driver Setup dialog, under the General tab, complete the following steps:
   a. In the Data Source Name field, enter any name.
   **Example**:
   `SiteMinder DB2 Wire Data Source`
   b. (Optional) In the Description field, enter a description of the DB2 wire protocol data source.
   c. In the IP Address field, enter the IP Address where the DB2 database is installed.
   d. In the Tcp Port field, enter the port number where DB2 is listening on the system.
   e. Click Test Connect.
   The connection is tested.

5. Click OK.
   The ODBC DB2 Wire Protocol Driver Setup dialog closes, the selections are saved, and the DB2 data source is created on a Windows System.

**Note:** You can now configure CA Single Sign-On to use the data source that you created.

## Create a DB2 Data Source on UNIX Systems

The CA Single Sign-on ODBC data sources are configured using a system_odbc.ini file, which you can create by renaming db2wire.ini, located in policy_server_home/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-on.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-on. The remaining attributes are specific to the driver.

Adding a DB2 Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the DB2 driver under [CA Single Sign-on Data Source].

Again, to configure a DB2 data source, you must first create a system_odbc.ini file in thepolicy_server_home/db directory. To do this, you need to rename db2wire.ini, located inpolicy_server_home/db, to system_odbc.ini.

**Note:** policy_server_home specifies the Policy Server installation path.

## Configure the DB2 Wire Protocol Driver

The following table contains configuration parameters for DB2 data sources. You can edit these parameters to configure data sources for separate key, audit log, session, and sample users databases.

| Parameter | Description | How to Edit |
|---|---|---|
| | Name of the data source. | |

| | | |
|---|---|---|
| Data Source Name | | Enter the data source name inside the square brackets. |
| Driver | Full path to the CA Single Sign-on DB2 Wire Protocol driver. | Replace “nete_ps_root” with the CA Single Sign-on installation directory. |
| Description | Description of the data source. | Enter any desired description. |
| Database | Name of the DB2 UDB database. | Replace “nete_database” with the name of the database configured on the DB2 UDB server. |
| LogonID | Username required for accessing the database. | Replace “uid” with the username of the DB2 UDB administrator. |
| Password | Password required for accessing the database. | Replace “pwd” with the password of the DB2 UDB administrator. |
| IPAddress | IP address or hostname of the DB2 UDB server. | Replace “nete_server_ip” with the IP address or the hostname of the DB2 UDB server. |
| TcpPort | TCP port number of the DB2 UDB server. | Replace the default value of 50000 with the actual TCP port number of the DB2 UDB server. |
| Package | The name of the package to process dynamic SQL. | Replace “nete_package” with the name of the package you want to create. |
| PackageOwner | (Optional) The AuthID assigned to the package. | Empty by default. This DB2 AuthID must have authority to execute all SQLs in the package. |
| GrantAuthid | The AuthID granted execute privileges for the package. | “PUBLIC” by default. Specify the desired AuthID if you wish to restrict the execute privileges for the package. |
| GrantExecute | Specifies whether to grant execute privileges to the AuthID listed in GrantAuthid. | Can be either 1 or 0. Set to 0 by default. |
| IsolationLevel | The method by which locks are acquired and released by the system. | CURSOR_STABILITY by default. |
| DynamicSections | The number of statements that the DB2 Wire Protocol driver package can prepare for a single user. | 100 by default. Enter the desired number of statements. |

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can access the CA Single Sign-On data in the policy store.

**Follow these steps:**

1. Open the Policy Server Management Console and click the Data tab.
2. Select the following value from the Storage list:

   `ODBC`
3. Select the following value from the Database list:

   `Policy Store`

4. Enter the name of the data source in the Data Source Information field.
   - (Windows) The entry must match the name that you entered in the Data Source Name field when you created the data source.
   - (UNIX) The entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure to enter the correct value.
5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
6. Specify the maximum number of database connections that are allocated to CA Single Sign-On.

   **Note:** We recommend retaining the 25 connection default for best performance.

7. Click Apply to save the settings.
8. Select the following value from the Database list:

   `Key Store`
9. Select the following value from the Storage list:

   `ODBC`
10. Select the following option:

    `Use the Policy Store database`
11. Select the following value from the Database list:

    `Audit Logs`
12. Select the following value from the Storage list:

    `ODBC`
13. Select the following option:

    `Use the Policy Store database`
14. Click Apply to save the settings.
15. Click Test Connection to verify that the Policy Server can access the policy store.
16. Click OK.
    The Policy Server is configured to use the database as a policy store, key store, and logging database.

## Set the CA Single Sign-On Super User Password

The default CA Single Sign-On administrator account is named **siteminder**. The account has maximum permissions.

Do not use the default super user for day-to-day operations. Use the default super user to:

- Access the Administrative UI for the first time.
- Manage CA Single Sign-On utilities for the first time.
- Create another administrator with super user permissions.

**Follow these steps:**

1. Copy the smreg utility to *siteminder_home*\bin.
   - *siteminder_home*
     Specifies the Policy Server installation path.

   > **Note:** The utility is at the top level of the Policy Server installation kit.

2. Run the following command:

   ```
   smreg -su password
   ```

   - *password*
     Specifies the password for the default administrator.

   The password has the following requirements:

   - The password must contain at least six (6) characters and cannot exceed 24 characters.
   - The password cannot include an ampersand (&) or an asterisk (*).
   - If the password contains a space, enclose the passphrase with quotation marks.

   > **Note:** If you are configuring an Oracle policy store, the password is case-sensitive. The password is not case-sensitive for all other policy stores.

3. Delete smreg from *siteminder_home*\bin. Deleting smreg prevents someone from changing the password without knowing the previous one.

The password for the default administrator account is set.

## Import the Policy Store Data Definitions

Importing the policy store data definitions defines the types of objects that can be created and stored in the policy store.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\xps\dd.
   - ***siteminder_home***
     Specifies the Policy Server installation path.
2. Run the following command:

   ```
   XPSDDInstall SmMaster.xdd
   ```

   - **XPSDDInstall**
     Imports the required data definitions.

## Import the Default Policy Store Objects

Importing the default policy store objects configures the policy store for use with the Administrative UI and the Policy Server.

Consider the following items:

- Be sure that you have write access to *siteminder_home*\bin. The import utility requires this permission to import the policy store objects.
  - **siteminder_home**
    Specifies the Policy Server installation path.
- Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges. For more information, see the release notes for your CA Single Sign-On component.

**Follow these steps:**

1. Open a command window and navigate to *siteminder_home*\db.
2. Import one of the following files:
   - To import smpolicy.xml, run the following command:

     ```
     XPSImport smpolicy.xml -npass
     ```

   - To import smpolicy–secure.xml, run the following command:

     ```
     XPSImport smpolicy-secure.xml -npass
     ```

     - **npass**
       Specifies that no passphrase is required. The default policy store objects do not contain encrypted data.

Both files include the default policy store objects. These objects include the default security settings in the default Agent Configuration Object (ACO) templates. The smpolicy–secure file provides more restrictive security settings. For more information, see Default Policy Store Objects Consideration (see page 213).

- To import Option Pack functionality, run the following command:

  ```
  XPSImport ampolicy.xml -npass
  ```

- To import federation functionality, run the following command:

  ```
  XPSImport fedpolicy-12.5.xml -npass
  ```

The policy store objects are imported.

**Note:** Importing ampolicy.xml makes available legacy federation and Web Service Variables functionality that is separately licensed from CA Single Sign-On. If you intend on using the latter functionality, contact your CA account representative for licensing information.

## Enable the Advanced Authentication Server

Enable the Advanced Authentication Server as part of configuring your Policy Server.

**Follow these steps:**

1. Start the Policy Server configuration wizard.
2. Perform one of the following steps:

   **On Windows:**
   Leave all the check boxes in the first screen of the wizard *cleared* and click Next.

   **On Linux:**
   Type 5 and press Enter.

3. Create the master encryption key for the Advanced Authentication Server.

   **Note**: If you are installing another (nth) Policy Server, use the same encryption key for the Advanced Authentication server that you used previously.

4. Complete the rest of the Policy Server configuration wizard.
   The Advanced Authentication Server is enabled.

## Prepare for the Administrative UI Registration

You use the default super user account (siteminder) to log into the Administrative UI for the first time. The initial login requires that you to register the Administrative UI with a Policy Server, which creates a trusted relationship between both components.

You prepare for the registration by using the XPSRegClient utility to supply the super user account name and password. The Policy Server uses these credentials to verify that the registration request is valid and that the trusted relationship can be established.

Consider the following items:

- The time from which you supply the credentials to when the initial Administrative UI login occurs is limited to 24 hours. If you do not plan on installing the Administrative UI within one day, complete the following steps before installing the Administrative UI.
- (UNIX) Be sure that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

```
XPSRegClient siteminder[:passphrase] -adminui-setup -t timeout -r retries -
c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

- **passphrase**
  Specifies the password for the default super user account (siteminder).

  **Note:** If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm one.

- **-adminui–setup**
  Specifies that the Administrative UI is being registered with a Policy Server for the first–time.

- **-t timeout**
  (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
  **Unit of measurement:** minutes
  **Default:** 240 (4 hours)
  **Minimum:** 15
  **Maximum:** 1440 (24 hours)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

  
  **Note:** Surround comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:***siteminder_home*\log
  *siteminder_home*
  Specifies the Policy Server installation path.

- **-e error_path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

3. Press Enter.
   XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

# Configure ODBC Databases as Key Store

This section explains the following Key Store configurations:



## Store Key Information in IBM DB2

**Contents**



### Gather Database Information

Configuring a single IBM DB2 database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Consider the following items:

- Information that is prefixed with a W represents a Windows requirement.
- Information that is prefixed with a U represents a UNIX requirement.

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store. You can use the IBM DB2 Information Worksheet to record your values.

- **Database instance name** —Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account** —Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.

- **Administrative password** —Determine the password for the Administrative account.
- **IP address**—Determine the IP address of the database host system.
- **Tcp port**—Determine the port on which the database is listening.
- (W) **Data source name** —Determine the name that is to identify the data source.
- (U) **Policy Server root** —Determine the explicit path to where the Policy Server is installed.
- (U) **Package** —Determine the name of the package that is to process dynamic SQL.
- (U) **Package owner**—Determine the AuthID assigned to the package. The AuthID must have the authority to execute all SQLs in the package.
- (U) **Grant AuthID**—If you want to restrict execute privileges for the package, determine the AuthID that is granted execute permissions for the package.
  **Default wire protocol setting:** Public
- (U) **Isolation level**—Determine the method by which the system acquires and releases locks.
  **Default wire protocol setting:** CURSOR_STABILTY
- (U) **Dynamic sections**—Determine the number of sections that the wire protocol driver package can prepare for a single user.
  **Default wire protocol setting:** 100

## Create the Key Store Schema

You create the CA Single Sign-On schema so that an IBM DB2 database can store key information.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*\db\tier2\DB2.
   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file and copy the contents to a text editor:

   `sm_db2_ps.sql`
4. Paste the contents into a query and execute the query.

   **Note:** For more information executing a query, see the IBM documentation.

   The key store schema is added to the database.

## Configure an IBM DB2 Data Source for CA Single Sign-On

If you are using ODBC, configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## Create a DB2 Data Source on Windows Systems

When using ODBC, you can create a DB2 data source for the DB2 wire protocol driver.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab and click Add.
3. Scroll down and select CA Single Sign-On DB2 Wire Protocol and click Finish.
4. In the ODBC DB2 Wire Protocol Driver Setup dialog, under the General tab, complete the following steps:
   a. In the Data Source Name field, enter any name.
   **Example**:
   `SiteMinder DB2 Wire Data Source`
   b. (Optional) In the Description field, enter a description of the DB2 wire protocol data source.
   c. In the IP Address field, enter the IP Address where the DB2 database is installed.
   d. In the Tcp Port field, enter the port number where DB2 is listening on the system.
   e. Click Test Connect.
   The connection is tested.
5. Click OK.
   The ODBC DB2 Wire Protocol Driver Setup dialog closes, the selections are saved, and the DB2 data source is created on a Windows System.

**Note:** You can now configure CA Single Sign-On to use the data source that you created.

## Create a DB2 Data Source on UNIX Systems

The CA Single Sign-on ODBC data sources are configured using a system_odbc.ini file, which you can create by renaming db2wire.ini, located in policy_server_home/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-on.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-on. The remaining attributes are specific to the driver.

Adding a DB2 Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the DB2 driver under [CA Single Sign-on Data Source].

Again, to configure a DB2 data source, you must first create a system_odbc.ini file in thepolicy_server_home/db directory. To do this, you need to rename db2wire.ini, located inpolicy_server_home/db, to system_odbc.ini.

**Note:** policy_server_home specifies the Policy Server installation path.

## Configure the DB2 Wire Protocol Driver

The following table contains configuration parameters for DB2 data sources. You can edit these parameters to configure data sources for separate key, audit log, session, and sample users databases.

| Parameter | Description | How to Edit |
|---|---|---|
| Data Source Name | Name of the data source. | Enter the data source name inside the square brackets. |
| Driver | Full path to the CA Single Sign-on DB2 Wire Protocol driver. | Replace "nete_ps_root" with the CA Single Sign-on installation directory. |
| Description | Description of the data source. | Enter any desired description. |
| Database | Name of the DB2 UDB database. | Replace "nete_database" with the name of the database configured on the DB2 UDB server. |

| | | |
|---|---|---|
| LogonID | Username required for accessing the database. | Replace “uid” with the username of the DB2 UDB administrator. |
| Password | Password required for accessing the database. | Replace “pwd” with the password of the DB2 UDB administrator. |
| IPAddress | IP address or hostname of the DB2 UDB server. | Replace “nete_server_ip” with the IP address or the hostname of the DB2 UDB server. |
| TcpPort | TCP port number of the DB2 UDB server. | Replace the default value of 50000 with the actual TCP port number of the DB2 UDB server. |
| Package | The name of the package to process dynamic SQL. | Replace “nete_package” with the name of the package you want to create. |
| PackageOwner | (Optional) The AuthID assigned to the package. | Empty by default. This DB2 AuthID must have authority to execute all SQLs in the package. |
| GrantAuthid | The AuthID granted execute privileges for the package. | “PUBLIC” by default. Specify the desired AuthID if you wish to restrict the execute privileges for the package. |
| GrantExecute | Specifies whether to grant execute privileges to the AuthID listed in GrantAuthid. | Can be either 1 or 0. Set to 0 by default. |
| IsolationLevel | The method by which locks are acquired and released by the system. | CURSOR_STABILITY by default. |
| DynamicSections | The number of statements that the DB2 Wire Protocol driver package can prepare for a single user. | 100 by default. Enter the desired number of statements. |

## Point the Policy Server to Database

You point the Policy Server to the database so the Policy Server can read and store key information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Key Store from the Database list and clear the Use Policy Store database check box.
   Data source settings become active.

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

6. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

7. Click Apply.
   The settings are saved.

8. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

9. Click OK.
   The Policy Server is configured to use the database as a key store

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Key Information in MySQL

**Contents**

## Before You Begin

Be sure that MySQL is installed using the default character set (Latin1). If MySQL was not installed using the default character set, reinstall MySQL before configuring the data store.

## Gather Database Information

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store:

- **Database host**—Identify the name of the database host system.
- **Database name**—Identify the name of the database instance that is to function as the policy store or data store.
- **Database port**—Identify the port on which the database is listening.
- **Administrator account**—Identify the login ID of an administrator account with permission to manage objects in the database.
- **Administrator password** —Identify the password for the administrator account.

## Create the Key Store Schema

You create the key store schema so the MySQL database can store key information.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to the following location:

   ```
   siteminder_home\db\tier2\MySQL.
   ```

   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file in a text editor:

   ```
   sm_mysql_ps.sql
   ```
4. Locate the following lines:

   ```
   DROP FUNCTION IF EXISTS `databaseName`.`getdate` $$
   CREATE FUNCTION `databaseName`.`getdate` () RETURNS DATE
   ```
5. Replace each instance of 'databaseName' with the name of the database functioning as the key store.
6. Copy the contents of the entire file.
7. Paste the file contents into a query and execute the query.
   The key store schema is created.

## Configure a MySQL Data Source for CA Single Sign-On

You configure a data source to let the Policy Server communicate with the CA Single Sign-On data store.

**Note**: If you are using MySQL 5.1.x, ensure that you assign the TRIGGER permission to the user name that is used to create the DSN.

## Create a MySQL Data Source on Windows

You create a MySQL data source for the MySQL wire protocol driver.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Do one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
3. Click System DSN.
4. Click Add.
5. Scroll down and select CA Single Sign-On MySQL Wire Protocol and click Finish.
6. Complete the following steps in the General tab:
   a. Enter a data source name in the Data Source Name field.
   **Example:**

   ```
   CA SiteMinder® MySQL Wire Data Source
   ```

   b. Enter the name of the MySQL database host system in the Host Name field.
   c. Enter the port on which the MySQL database is listening in the Port Number field.
   d. Enter the name of the MySQL database in the Database Name field.
7. Click Test Connect.

8. Click OK.
   The data source is created and appears in the System Data Sources list.

**Note:** You can now point the Policy Server to the CA Single Sign-On data store.

## Create a MySQL Data Source on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming mysqlwire.ini to system_odbc.ini. The mysqlwire.ini file is located in *siteminder_home*/db.

- **siteminder_home**
  Specifies the Policy Server installation path.

This system_odbc.ini file contains all of the names of the available ODBC data sources and the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add other data sources to this file, such as defining other ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** The value of the first line of data source entry is required when you configure the database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver that is loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

Adding a MySQL Server Data source includes:

- A new data source name in the [ODBC Data Sources] section of the file.
- A section that describes the data source using the same name as the data source.

Update the system_odbc.ini file when creating a new service name. You have entries for the MySQL driver under [CA Single Sign-On Data Source].

Again, to configure a MySQL Server data source, you create the system_odbc.ini file by renaming mysqlwire.ini to system_odbc.ini.

## Create the MySQL Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following line under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 MySQL Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSmysql27.so
   Description=DataDirect 7.1 MySQL Wire Protocol
   Database=database_nameHostName=host_nameLogonID=root_userPassword=root_user_pass
   wordPortNumber=mysql_port
   ```

- *nete_ps_root*
  Specifies the Policy Server installation path. Enter this value as an explicit path, rather than one with an environment variable.
  **Example:** /export/smuser/siteminder

- *database_name*
  Specifies the name of the MySQL database that is to function as the data store.

- *host_name*
  Specifies the name of the MySQL database host system.

- *root_user*
  Specifies the login ID of the MySQL root user.

- *root_user_password*
  Specifies the password for the MySQL root user.

- *mysql_port*
  Specifies the port on which the MySQL database is listening.

4. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to Database

You point the Policy Server to the database so the Policy Server can read and store key information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Key Store from the Database list and clear the Use Policy Store database check box.
   Data source settings become active.

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

6. Specify the maximum number of database connections allocated to CA Single Sign-On.

**Note:** We recommend retaining the default for best performance.

7. Click Apply.
   The settings are saved.

8. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

9. Click OK.
   The Policy Server is configured to use the database as a key store

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Key Information in Oracle

**Contents**



### Gather Database Information

Configuring a single Oracle database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Information prefixed with (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. This information is required when configuring Oracle data source for UNIX.

## Required Information

Gather the following required information before configuring a supported Oracle or Oracle RAC database as a policy store or any other type of CA Single Sign-On data store:

- (U) **Policy Server installation path**—Identify the explicit path to where the Policy Server is installed.
- **Data source**—Determine the name you will use to identify the Oracle data source.
  **Example:** SM Oracle Server Wire DS.
- **Database administrative account**—Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.

  **Note:** Ensure the administrative account does not have the DB role. Audit-based reports will not return correct results if the administrative account has the DB role.
- **Database administrative Password**—Determine the password for the Administrative account.

## Oracle Database Information

Gather the following information only if you are configuring a supported Oracle database as a policy store or any other type of CA Single Sign-On data store:

- **Oracle machine name**—Determine the name of the machine on which the Oracle database is installed.
- **Oracle instance service name**—Determine the service name of the database instance to which you will connect. The tnsnames.ora file specifies service names.
- **Oracle port number**—Determine the port number on which the Oracle database is listening.

## Oracle RAC Database (without SCAN) Information

Gather the following information if you are configuring a supported Oracle RAC database (without SCAN functionality configured) as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name—**Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
(Description =
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername1)(PORT=1521
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername2)(PORT=1521)
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername3)(PORT=1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA=
(SERVER = DEDICATED)
(SERVER_NAME = SMDB))
)
```

- **Oracle RAC node service names**—Determine the service names for each node in the system.

- **Oracle RAC node IP addresses**—Determine the IP Address of each node in the Oracle RAC system.

  **Note:** If you are using Oracle RAC 10g, determine the virtual IP address of each node in the system.

- **Oracle RAC node port numbers**—Determine the port number for each node in the Oracle RAC system.

## Oracle RAC Database (Using SCAN) Information

The Oracle RAC Single Client Access Name (SCAN) feature provides a single name for clients to access any Oracle Database running in a cluster.

Gather the following information if you are configuring an Oracle RAC database with SCAN functionality as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name**—Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SMDB)
    )
  )
```

- **Oracle RAC SCAN Address**—Determine the FQDN of the Oracle RAC system SCAN.

- **Oracle RAC SCAN port number**—Determine the port number for the Oracle RAC system SCAN.

## Create the Key Store Schema

You create the key store schema so the Oracle database can store key information.

**To create the CA Single Sign-On schema**

1. Log into Oracle with sqlplus or some other Oracle utility as the user who administers the Policy Server database information.

   **Note:** We recommend that you do not create CA Single Sign-On schema with the SYS or SYSTEM users. If necessary, create an Oracle user, such as SMOWNER, and create the schema with that user.

2. Import the following script:
   $NETE_PS_ROOT/db/sql/sm_oracle_ps.sql
   **Note:** Environment variables may not function in Oracle's SQL utility. If you experience problems importing the script using the utility, specify an explicit path.

3. Create a table space for the key store.
4. Create a user with the following privileges to manage the table space in the database:

- Connect
- Resource
- Unlimited Tablespace

## Configure an Oracle Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source for the Oracle wire protocol driver.

## Create an Oracle Data Source on Windows

Create an ODBC data source for an Oracle database.

**Follow these steps:**

1. Do one of the following:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears
3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.
4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.
5. Enter the machine name where the Oracle database is installed in the Host Name field.
6. Enter the port number where the Oracle database is listening on the machine in the Port Number field.

7. Enter the name of the Oracle instance to which you want to connect in the SID field.

   **Note:** The service name is specified in the tnsnames.ora file. The SID is the system identifier for the database instance. The tnsnames.ora file contains service names and details that Oracle uses to identify and connect to Oracle instances.

   **Example:** if the tnsnames.ora file contains the following entry for an Oracle instance, you enter instance1 in the SID field:

   ```
   instance1 =
       (Description=
       (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
       (Connect_DATA_ = (SID = SIDofinstance1))
       )
   ```

8. Click Test Connection.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The Oracle data source is configured for the wire protocol driver.

## Create an Oracle RAC (no SCAN) Data Source on Windows

Create an ODBC data source for an Oracle RAC database that does not use the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

**Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the IP Address of the first node in the Oracle RAC system in the Host field.
   Oracle RAC 10g: Enter the virtual IP Address.

6. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains 3 nodes:

```
SMDB=
    (Description =
(ADDRESS = (Protocol = TCP)(HOST = nete_servername1)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername2)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername3)(PORT = 1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = SMDB)
)
```

7. Click the Failover tab.
   Failover settings appear.

8. Specify the host name or virtual IP Address, port number, and service name for the remaining Oracle RAC nodes in the environment in the Alternate Servers field.

**Note:** The ServiceName is the service name for the entire Oracle RAC system.

9. Specify the AlternateServers to provide connection failover to the other Oracle nodes if the primary server is not accepting connections. The entry should have the following format:
   (HostName=nete_servername2:PortNumber=1521:ServiceName=nete_servicename[,...])

10. Select LoadBalancing.

11. Click OK
    The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle RAC SCAN Data Source on Windows

Create an ODBC data source for an Oracle RAC database that uses the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.

   - If you are using a supported 64–bit Windows operating system:

a. Navigate to the C:\Windows\SysWOW64.

b. Double–click odbcad32.exe

The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
The Create New Data Source dialog appears.

3. Select CA CA Single Sign-on Oracle Wire Protocol, and click Finish.
The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

> **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the FQDN or IP Address of the SCAN in the Host field.

6. Enter the port number of the SCAN in the Port Number field.

7. Enter the service name for the entire Oracle RAC system in the Service Name field.
**Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains the SCAN:

```
SMDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.rac.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORCL)
    )
  )
```

8. Click OK
The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle Data Source on UNIX Systems

You configure the names of available ODBC data sources and the attributes that are associated with these data sources in the system_odbc.ini file.

**To create the system_odbc.ini file:**

1. Navigate to *policy_server_installation*/db

2. Rename oraclewire.ini to "system_odbc.ini".

Customize the system_odbc.ini file for each site. You can also add more data sources to this file, such as defining extra ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of the data source entry ([CA Single Sign-On Data Source]), take note of the change. This value is required to configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

**To add an Oracle Data source**:

1. Define a new data source name in the [ODBC Data Sources] section of the file.
2. Add a section that describes the data source using the same name as the data source.

To create a service name or use a different driver, edit the system_odbc.ini file. Entries for the SQL Server or Oracle drivers belong under [CA Single Sign-On Data Source].

## Configure the Oracle Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so (http://nsora27.so/)Description=DataDirect
    7.1 Oracle Wire Protocol
   LoginID=uidPassword=pwdHostName=host_namePortNumber=1521
   SID=server_idCatalogOptions=0
   ProcedureResults=0
   EnableDisableParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation.

   - *uid*
     Specifies the user name of the database account that has full access rights to the database.

   - *pwd*
     Specifies the password for the database account that has full access rights to the database.

   - *host_name*
     Specifies the name of the Oracle database host system.
     *server_id*
     Specifies the Oracle instance service name (SID). The SID is the system identifier for the database instance.

   **Example:** In the following sample tnsnames.ora file, the value instance1 is the SID

   ```
   instance1 =
   (Description =
   (ADDRESS = (Protocol = TCP)(Host = myhost)(Port = 1521)
   (CONNECT_DATA = (SID = instance1))
   )
   ```

3. Save the file.

The Oracle wire protocol driver is configured.

## Configure the Oracle Wire Protocol Driver for Oracle RAC without SCAN

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.
   - Add ServiceName=*nete_servicename*
   - Add AlternateServers=

- Add Loadbalancing=1
- Remove or comment SID=nete_serverid

The modified text for the data source should appear as follows:

```
Driver=nete_ps_root/odbc/lib/NSora27.so
Description=DataDirect 7.1 Oracle Wire Protocol
Logon=uidPassword=pwdHostName=server_name1PortNumber=1521
ServiceName=service_nameCatalogOptions=0
ProcedureRetResults=0
EnableDescribeParam=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
AlternateServers=
LoadBalancing=1
```

- *nete_ps_root*
  Specifies an explicit path to the directory where Policy Server is installed.
- *uid*
  Specifies the user name of the database account that has full access rights to the database.
- *pwd*
  Specifies the password for the database account that has full access rights to the database.
- *server_name1*
  Specifies the IP Address of the first Oracle RAC node.
  (Oracle 10g) Specifies the virtual IP Address of the first Oracle RAC node.
- *service_name*
  Specifies the Oracle RAC system service name for the entire RAC system.
- **AlternateServers=**
  If the primary server is not accepting connections, specifies the connection failover to the other Oracle nodes.
  **Example:** (HostName=nete_servername2:PortNumber=1521:
  ServiceName=nete_servicename[,...])
- **LoadBalancing=1**
  Turns on client load balancing, which helps to distribute new connections to keep RAC nodes from being overwhelmed with connection requests. When enabled, the order in which primary and alternate database servers are accessed is random.

3. Save the file.
   The Oracle wire protocol driver is configured.

## Point the Policy Server to Database

You point the Policy Server to the database so the Policy Server can read and store key information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Key Store from the Database list and clear the Use Policy Store database check box.
   Data source settings become active.

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

6. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

7. Click Apply.
   The settings are saved.

8. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

9. Click OK.
   The Policy Server is configured to use the database as a key store

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

**Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Store Key Information in PostgreSQL

**Contents**



To configure a PostgreSQL Server database as a standalone key store, complete the following procedures:

1. Gather database information.
2. Create the key store schema.
3. Configure a PostgreSQL Server data source for CA Single Sign-On.
4. Point the Policy Server to the database.
5. Restart the Policy Server.

## Create the Key Store Schema

You create the key store schema so the SQL Server database can store key information.

**To create the key store schema**

1. Open sm_postgresql_ps.sql in a text editor and copy the contents of the entire file.
2. Open a SQL client, such as psql, and log in as the who administers the Policy Server database.
3. Select the database instance from the database list.
4. Paste the schema from sm_postgresql_ps.sql into the query.
5. Execute the query.
   The CA Single Sign-On key store schema is created in the database.

## Point the Policy Server to Database

You point the Policy Server to the database so the Policy Server can read and store key information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Key Store from the Database list and clear the Use Policy Store database check box.
   Data source settings become active.

4. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

6. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

7. Click Apply.
   The settings are saved.

8. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

9. Click OK.
   The Policy Server is configured to use the database as a key store

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

**Note**: On UNIX, execute the stop-ps and start-ps commands respectively to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Store Key Information in SQL Server

**Contents**



## Gather Database Information

Configuring a single SQL Server database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

**Note:** Information prefixed with (W) indicates that the information is only required if the Policy Server is installed on a Windows system; (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. Different information is required when configuring the SQL Server data source.

- **Database instance name**
  Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account name and password**
  Determine the user name and password of an account with privileges to create, read, modify, and delete objects in the database.
- **(W) Data source name**
  Determine the name you will use to identify the data source.
  **Example:** SM SQL Server Wire DS.
- **(W) SQL Server name**
  Determine the name of the SQL Server database that contains the instance that is to function as the policy store.
- **(U) Policy Server root**
  Determine the explicit path to where the Policy Server is installed.
- **(U) IP Address**
  Determine the IP Address of the SQL Server database.

## Create the Key Store Schema

You create the key store schema so the SQL Server database can store key information.

**To create the key store schema**

1. Open sm_mssql_ps.sql in a text editor and copy the contents of the entire file.

2. Start the Query Analyzer and log in as the who administers the Policy Server database.
3. Select the database instance from the database list.
4. Paste the schema from sm_mssql_ps.sql into the query.
5. Execute the query.
   The CA Single Sign-On key store schema is created in the database.

## Configure a SQL Server Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## SQL Server Authentication Mode Considerations

CA Single Sign-On data sources do not support Windows authentication. Configure the CA Single Sign-On data source with the credentials of a user that is stored in the database.

**Note:** For more information about SQL Server authentication modes, see the vendor–specific documentation.

## Create a SQL Server Data Source on Windows

ODBC requires that you configure a data source for the SQL Server wire protocol.

**Note:** This procedure only applies if the Policy Server is installed on a Windows System.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab.
   System data source settings appear.

3. Click Add.
   The Create New Data Source dialog appears.

4. Select CA Single Sign-On SQL Server Wire Protocol and click Finish.
   The ODBC SQL Server Wire Protocol Driver Setup dialog appears.

5. Enter the data source name in the Data Source Name field.
   **Example:** CA Single Sign-On Data Source.
   **Note:** Take note of your data source name. This information is required as you configure your database as a policy store.

6. Enter the name of the SQL Server host system in the Server field.

7. Enter the database name in the Database Name field.

8. Click Test.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The SQL Server data source is configured and appears in the System Data Sources list.

## Create a SQL Server Data Sources on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of data source entry, which is [CA Single Sign-On Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-On. The remaining attributes are specific to the driver.

Adding a MS SQL Server Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the Oracle or SQL drivers under [CA Single Sign-On Data Source].

Again, to configure a MS SQL Server data source, you must first create a system_odbc.ini file in the *policy_server_installation*/db directory. To do this, you need to rename sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini.

## Configure the SQL Server Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following under [ODBC Data Sources]:

```
SiteMinder Data Source=DataDirect 7.1 SQL Server Wire Protocol
```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

```
Driver=nete_ps_root/odbc/lib/NSsqls27.so
Description=DataDirect 7.1 SQL Server Wire Protocol
Database=SiteMinder DataAddress=host_ip, 1433
QuotedId=No
AnsiNPW=No
```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation, rather than a path with an environment variable.
     **Example:** export/smuser/siteminder

   - *SiteMinder Data*
     Specifies the SQL Server database instance name.

   - *host_ip*
     Specifies the IP Address of the SQL Server database.

   - **1433**
     Represents the default listening port for SQL Server.

4. If you are using Microsoft SQL Server 2008 to function as any CA Single Sign-On store, edit the [ODBC] section as follows:

```
TraceFile=nete_ps_root/db/odbctrace.out
TraceDll=nete_ps_root/odbc/lib/NStrc27.so
InstallDir=nete_ps_root/odbc
```

   - *nete_ps_root*
     Specifies the explicit path to the Policy Server installation directory. This path cannot contain an environment variable.

5. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to Database

You point the Policy Server to the database so the Policy Server can read and store key information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Key Store from the Database list and clear the Use Policy Store database check box.
   Data source settings become active.

4. Enter the name of the data source in the Data Source Information field.

- (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.
- (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

5. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
6. Specify the maximum number of database connections allocated to CA Single Sign-On.

**Note:** We recommend retaining the default for best performance.

7. Click Apply.
   The settings are saved.
8. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.
9. Click OK.
   The Policy Server is configured to use the database as a key store

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Configure ODBC Databases as Session Store

This section explains the following Session Store configurations:

- Store Session Information in IBM DB2 (see page 317)
- Store Session Information in MySQL (see page 319)
- Store Session Information in Oracle (see page 325)
- Store Session Information in PostgreSQL (see page 337)

# Store Session Information in IBM DB2

**Contents**



## Gather Database Information

Configuring a single IBM DB2 database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Consider the following items:

- Information that is prefixed with a W represents a Windows requirement.
- Information that is prefixed with a U represents a UNIX requirement.

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store. You can use the IBM DB2 Information Worksheet to record your values.

- **Database instance name** —Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account** —Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.
- **Administrative password** —Determine the password for the Administrative account.
- **IP address**—Determine the IP address of the database host system.
- **Tcp port**—Determine the port on which the database is listening.
- (W) **Data source name** —Determine the name that is to identify the data source.
- (U) **Policy Server root** —Determine the explicit path to where the Policy Server is installed.
- (U) **Package** —Determine the name of the package that is to process dynamic SQL.
- (U) **Package owner**—Determine the AuthID assigned to the package. The AuthID must have the authority to execute all SQLs in the package.

- (U) **Grant AuthID**—If you want to restrict execute privileges for the package, determine the AuthID that is granted execute permissions for the package.
  **Default wire protocol setting:** Public

- (U) **Isolation level**—Determine the method by which the system acquires and releases locks.
  **Default wire protocol setting:** CURSOR_STABILTY

- (U) **Dynamic sections**—Determine the number of sections that the wire protocol driver package can prepare for a single user.
  **Default wire protocol setting:** 100

## Create the Session Store Schema

You create the CA Single Sign-On schema so that an IBM DB2 database can store session information.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*\db\tier2\DB2.
   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file and copy the contents to a text editor:

   ```
   sm_db2_ss.sql
   ```
4. Paste the contents into a query and execute the query.

   **Note:** For more information executing a query, see the IBM documentation.

   The session store schema is added to the database.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store session information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.
2. Select Session Server from the Database list.
   Data source settings become active.
3. Enter the name of the data source in the Data Source Information field.

- (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.
- (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

4. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
5. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

6. Click Apply.
   The settings are saved.
7. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.
8. Click OK.
   The Policy Server is configured to use the database as a session store.

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Session Information in MySQL

**Contents**

- Restart the Policy Server (see page 325)

## Before You Begin

Be sure that MySQL is installed using the default character set (Latin1). If MySQL was not installed using the default character set, reinstall MySQL before configuring the data store.

## Gather Database Information

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store:

- **Database host**—Identify the name of the database host system.
- **Database name**—Identify the name of the database instance that is to function as the policy store or data store.
- **Database port**—Identify the port on which the database is listening.
- **Administrator account**—Identify the login ID of an administrator account with permission to manage objects in the database.
- **Administrator password** —Identify the password for the administrator account.

## Create the Session Store Schema

You create the session store schema so the MySQL database can store the session information.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to the following location:

   ```
   siteminder_home\db\tier2\MySQL.
   ```

   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file in a text editor:

   ```
   sm_mysql_ss.sql
   ```
4. Locate the following lines:

   ```
   DROP FUNCTION IF EXISTS `databaseName`.`getdate` $$
   CREATE FUNCTION `databaseName`.`getdate` () RETURNS DATE
   ```
5. Replace each instance of 'databaseName' with the name of the database functioning as the session store.
6. Copy the contents of the entire file.
7. Paste the file contents into a query and execute the query.
   The session store schema is created.

## Configure a MySQL Data Source for CA Single Sign-On

You configure a data source to let the Policy Server communicate with the CA Single Sign-On data store.

**Note**: If you are using MySQL 5.1.x, ensure that you assign the TRIGGER permission to the user name that is used to create the DSN.

## Create a MySQL Data Source on Windows

You create a MySQL data source for the MySQL wire protocol driver.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Do one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
3. Click System DSN.
4. Click Add.
5. Scroll down and select CA Single Sign-On MySQL Wire Protocol and click Finish.
6. Complete the following steps in the General tab:
   a. Enter a data source name in the Data Source Name field.
   **Example:**

   ```
   CA SiteMinder® MySQL Wire Data Source
   ```

   b. Enter the name of the MySQL database host system in the Host Name field.
   c. Enter the port on which the MySQL database is listening in the Port Number field.
   d. Enter the name of the MySQL database in the Database Name field.
7. Click Test Connect.

8. Click OK.
   The data source is created and appears in the System Data Sources list.

**Note:** You can now point the Policy Server to the CA Single Sign-On data store.

## Create a MySQL Data Source on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming mysqlwire.ini to system_odbc.ini. The mysqlwire.ini file is located in *siteminder_home*/db.

- **siteminder_home**
  Specifies the Policy Server installation path.

This system_odbc.ini file contains all of the names of the available ODBC data sources and the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add other data sources to this file, such as defining other ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** The value of the first line of data source entry is required when you configure the database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver that is loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

Adding a MySQL Server Data source includes:

- A new data source name in the [ODBC Data Sources] section of the file.
- A section that describes the data source using the same name as the data source.

Update the system_odbc.ini file when creating a new service name. You have entries for the MySQL driver under [CA Single Sign-On Data Source].

Again, to configure a MySQL Server data source, you create the system_odbc.ini file by renaming mysqlwire.ini to system_odbc.ini.

## Create the MySQL Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following line under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 MySQL Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSmysql27.so
   Description=DataDirect 7.1 MySQL Wire Protocol
   Database=database_nameHostName=host_nameLogonID=root_userPassword=root_user_pass
   wordPortNumber=mysql_port
   ```

- *nete_ps_root*
  Specifies the Policy Server installation path. Enter this value as an explicit path, rather than one with an environment variable.
  **Example:** /export/smuser/siteminder

- *database_name*
  Specifies the name of the MySQL database that is to function as the data store.

- *host_name*
  Specifies the name of the MySQL database host system.

- *root_user*
  Specifies the login ID of the MySQL root user.

- *root_user_password*
  Specifies the password for the MySQL root user.

- *mysql_port*
  Specifies the port on which the MySQL database is listening.

4. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store session information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select Session Server from the Database list.
   Data source settings become active.

3. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

4. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

5. Specify the maximum number of database connections allocated to CA Single Sign-On.

**Note:** We recommend retaining the default for best performance.

6. Click Apply.
   The settings are saved.

7. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

8. Click OK.
   The Policy Server is configured to use the database as a session store.

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Session Information in Oracle

**Contents**



### Gather Database Information

Configuring a single Oracle database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Information prefixed with (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. This information is required when configuring Oracle data source for UNIX.

### Required Information

Gather the following required information before configuring a supported Oracle or Oracle RAC database as a policy store or any other type of CA Single Sign-On data store:

- (U) **Policy Server installation path**—Identify the explicit path to where the Policy Server is installed.
- **Data source**—Determine the name you will use to identify the Oracle data source.
  **Example:** SM Oracle Server Wire DS.
- **Database administrative account**—Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.

  **Note:** Ensure the administrative account does not have the DB role. Audit-based reports will not return correct results if the administrative account has the DB role.

- **Database administrative Password**—Determine the password for the Administrative account.

## Oracle Database Information

Gather the following information only if you are configuring a supported Oracle database as a policy store or any other type of CA Single Sign-On data store:

- **Oracle machine name**—Determine the name of the machine on which the Oracle database is installed.
- **Oracle instance service name**—Determine the service name of the database instance to which you will connect. The tnsnames.ora file specifies service names.
- **Oracle port number**—Determine the port number on which the Oracle database is listening.

## Oracle RAC Database (without SCAN) Information

Gather the following information if you are configuring a supported Oracle RAC database (without SCAN functionality configured) as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name—**Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
(Description =
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername1)(PORT=1521
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername2)(PORT=1521)
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername3)(PORT=1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA=
(SERVER = DEDICATED)
(SERVER_NAME = SMDB))
)
```

- **Oracle RAC node service names**—Determine the service names for each node in the system.
- **Oracle RAC node IP addresses**—Determine the IP Address of each node in the Oracle RAC system.

  **Note:** If you are using Oracle RAC 10g, determine the virtual IP address of each node in the system.

- **Oracle RAC node port numbers**—Determine the port number for each node in the Oracle RAC system.

## Oracle RAC Database (Using SCAN) Information

The Oracle RAC Single Client Access Name (SCAN) feature provides a single name for clients to access any Oracle Database running in a cluster.

Gather the following information if you are configuring an Oracle RAC database with SCAN functionality as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name**—Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SMDB)
    )
  )
```

- **Oracle RAC SCAN Address**—Determine the FQDN of the Oracle RAC system SCAN.
- **Oracle RAC SCAN port number**—Determine the port number for the Oracle RAC system SCAN.

## Create the Session Store Schema

Create the session store schema so the Oracle database can store the session information.

**Follow these steps:**

1. Log in to Oracle as the user who administers the database information. Log in with an Oracle utility, such as sqlplus.

   **Note:** We recommend that you do not create the schema with the SYS or SYSTEM users. If necessary, create an Oracle user, such as SMOWNER, and create the schema with that user.

2. To store Unicode characters, confirm that the character set for the Oracle database is set correctly. If you plan to use only English characters, skip this step.

   a. To find the character set, use the following query:
   SELECT value$ FROM sys.props$ WHERE name = 'NLS_CHARACTERSET' ;

   b. Verify that the character set is AL32UTF8 or UTF8 before importing the schema.

3. Import the following script:
   $NETE_PS_ROOT/db/sql/sm_oracle_ss.sql
   **Note:** Some Oracle SQL utilities have problems with environment variables. If you experience problems importing the script using the utility, specify an explicit path.

4. Create a table space for the session store.

5. Create a user with the following privileges to manage the table space in the database:

- Connect
- Resource
- Unlimited Tablespace

## Configure an Oracle Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source for the Oracle wire protocol driver.

## Create an Oracle Data Source on Windows

Create an ODBC data source for an Oracle database.

**Follow these steps:**

1. Do one of the following:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the *install_home*\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

**Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the machine name where the Oracle database is installed in the Host Name field.

6. Enter the port number where the Oracle database is listening on the machine in the Port Number field.

7. Enter the name of the Oracle instance to which you want to connect in the SID field.

   **Note:** The service name is specified in the tnsnames.ora file. The SID is the system identifier for the database instance. The tnsnames.ora file contains service names and details that Oracle uses to identify and connect to Oracle instances.

   **Example:** if the tnsnames.ora file contains the following entry for an Oracle instance, you enter instance1 in the SID field:

   ```
   instance1 =
       (Description=
       (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
       (Connect_DATA_ = (SID = SIDofinstance1))
       )
   ```

8. Click Test Connection.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The Oracle data source is configured for the wire protocol driver.

## Create an Oracle RAC (no SCAN) Data Source on Windows

Create an ODBC data source for an Oracle RAC database that does not use the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.

   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the C:\Windows\SysWOW64.

     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the IP Address of the first node in the Oracle RAC system in the Host field.
   Oracle RAC 10g: Enter the virtual IP Address.

6. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains 3 nodes:

```
SMDB=
    (Description =
(ADDRESS = (Protocol = TCP)(HOST = nete_servername1)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername2)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername3)(PORT = 1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = SMDB)
)
```

7. Click the Failover tab.
   Failover settings appear.

8. Specify the host name or virtual IP Address, port number, and service name for the remaining Oracle RAC nodes in the environment in the Alternate Servers field.

   **Note:** The ServiceName is the service name for the entire Oracle RAC system.

9. Specify the AlternateServers to provide connection failover to the other Oracle nodes if the primary server is not accepting connections. The entry should have the following format:
   (HostName=nete_servername2:PortNumber=1521:ServiceName=nete_servicename[,...])

10. Select LoadBalancing.

11. Click OK
    The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle RAC SCAN Data Source on Windows

Create an ODBC data source for an Oracle RAC database that uses the SCAN feature.

**Follow these steps:**

1. Do one of the following:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.
3. Select CA CA Single Sign-on Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.
4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.
5. Enter the FQDN or IP Address of the SCAN in the Host field.
6. Enter the port number of the SCAN in the Port Number field.
7. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains the SCAN:

```
SMDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.rac.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORCL)
    )
  )
```

8. Click OK
   The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle Data Source on UNIX Systems

You configure the names of available ODBC data sources and the attributes that are associated with these data sources in the system_odbc.ini file.

**To create the system_odbc.ini file:**

1. Navigate to *policy_server_installation*/db
2. Rename oraclewire.ini to "system_odbc.ini".

Customize the system_odbc.ini file for each site. You can also add more data sources to this file, such as defining extra ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of the data source entry ([CA Single Sign-On Data Source]), take note of the change. This value is required to configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

**To add an Oracle Data source**:

1. Define a new data source name in the [ODBC Data Sources] section of the file.
2. Add a section that describes the data source using the same name as the data source.

To create a service name or use a different driver, edit the system_odbc.ini file. Entries for the SQL Server or Oracle drivers belong under [CA Single Sign-On Data Source].

## Configure the Oracle Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.

- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so (http://nsora27.so/)Description=DataDirect
    7.1 Oracle Wire Protocol
   LoginID=uidPassword=pwdHostName=host_namePortNumber=1521
   SID=server_idCatalogOptions=0
   ProcedureResults=0
   EnableDisableParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation.

   - *uid*
     Specifies the user name of the database account that has full access rights to the database.

   - *pwd*
     Specifies the password for the database account that has full access rights to the database.

   - *host_name*
     Specifies the name of the Oracle database host system.
     *server_id*
     Specifies the Oracle instance service name (SID). The SID is the system identifier for the database instance.

   **Example:** In the following sample tnsnames.ora file, the value instance1 is the SID

```
instance1 =
(Description =
(ADDRESS = (Protocol = TCP)(Host = myhost)(Port = 1521)
(CONNECT_DATA = (SID = instance1))
)
```

3. Save the file.

The Oracle wire protocol driver is configured.

## Configure the Oracle Wire Protocol Driver for Oracle RAC without SCAN

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   - Add ServiceName=*nete_servicename*
   - Add AlternateServers=
   - Add Loadbalancing=1
   - Remove or comment SID=nete_serverid

   The modified text for the data source should appear as follows:

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so
   Description=DataDirect 7.1 Oracle Wire Protocol
   Logon=uidPassword=pwdHostName=server_name1PortNumber=1521
   ServiceName=service_nameCatalogOptions=0
   ProcedureRetResults=0
   EnableDescribeParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   AlternateServers=
   LoadBalancing=1
   ```

   - *nete_ps_root*
     Specifies an explicit path to the directory where Policy Server is installed.
   - *uid*
     Specifies the user name of the database account that has full access rights to the database.
   - *pwd*
     Specifies the password for the database account that has full access rights to the database.
   - *server_name1*
     Specifies the IP Address of the first Oracle RAC node.
     (Oracle 10g) Specifies the virtual IP Address of the first Oracle RAC node.
   - *service_name*
     Specifies the Oracle RAC system service name for the entire RAC system.
   - **AlternateServers=**
     If the primary server is not accepting connections, specifies the connection failover to the other Oracle nodes.
     **Example:** (HostName=nete_servername2:PortNumber=1521:
     ServiceName=nete_servicename[,...])
   - **LoadBalancing=1**
     Turns on client load balancing, which helps to distribute new connections to keep RAC nodes from being overwhelmed with connection requests. When enabled, the order in which primary and alternate database servers are accessed is random.

3. Save the file.
   The Oracle wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store session information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select Session Server from the Database list.
   Data source settings become active.

3. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

4. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

5. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

6. Click Apply.
   The settings are saved.

7. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

8. Click OK.
   The Policy Server is configured to use the database as a session store.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Session Information in PostgreSQL

**Contents**



To configure a PostgreSQL Server database as a standalone session store, complete the following procedures:

1. Gather database information.
2. Create the session store schema.
3. Configure a PostgreSQL Server data source for CA Single Sign-On.
4. Point the Policy Server to the database.
5. Restart the Policy Server.

### Create the Session Store Schema

You create the session store schema so the SQL Server database can store and read session information.

**To create the session store schema**

1. Open sm_postgresql_ss.sql in a text editor and copy the contents of the entire file.
2. Use an SQL client, such as psql, and log in as the who administers the Policy Server database.
3. Select the database instance from the database list.
4. Paste the schema into the query.
5. Execute the query.
   The session store schema is created in the database.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store session information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select Session Server from the Database list.
   Data source settings become active.

3. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

4. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

5. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

6. Click Apply.
   The settings are saved.

7. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

8. Click OK.
   The Policy Server is configured to use the database as a session store.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands respectively to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Session Information in SQL Server

**Contents**



### Gather Database Information

Configuring a single SQL Server database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

**Note:** Information prefixed with (W) indicates that the information is only required if the Policy Server is installed on a Windows system; (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. Different information is required when configuring the SQL Server data source.

- **Database instance name**
  Determine the name of the database instance that is to function as the policy store or data store.

- **Administrative account name and password**
  Determine the user name and password of an account with privileges to create, read, modify, and delete objects in the database.

- **(W) Data source name**
  Determine the name you will use to identify the data source.
  **Example:** SM SQL Server Wire DS.

- **(W) SQL Server name**
  Determine the name of the SQL Server database that contains the instance that is to function as the policy store.

- **(U) Policy Server root**
  Determine the explicit path to where the Policy Server is installed.

- **(U) IP Address**
  Determine the IP Address of the SQL Server database.

## Create the Session Store Schema

You create the session store schema so the SQL Server database can store and read session information.

**To create the session store schema**

1. Do one of the following:
   - If you are going to store Unicode characters in the session store, open sm_mssql_ss.sql.unicode in a text editor and copy the contents of the entire file.
   - If you are not going to store Unicode characters in the session store, open sm_mssql_ss.sql in a text editor and copy the contents of the entire file.
2. Start the Query Analyzer and log in as the who administers the Policy Server database.
3. Select the database instance from the database list.
4. Paste the schema into the query.
5. Execute the query.
   The session store schema is created in the database.

## Configure a SQL Server Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## SQL Server Authentication Mode Considerations

CA Single Sign-On data sources do not support Windows authentication. Configure the CA Single Sign-On data source with the credentials of a user that is stored in the database.

**Note:** For more information about SQL Server authentication modes, see the vendor–specific documentation.

## Create a SQL Server Data Source on Windows

ODBC requires that you configure a data source for the SQL Server wire protocol.

**Note:** This procedure only applies if the Policy Server is installed on a Windows System.

**Follow these steps:**

1. Complete one of the following steps:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab.
   System data source settings appear.

3. Click Add.
   The Create New Data Source dialog appears.

4. Select CA Single Sign-On SQL Server Wire Protocol and click Finish.
   The ODBC SQL Server Wire Protocol Driver Setup dialog appears.

5. Enter the data source name in the Data Source Name field.
   **Example:** CA Single Sign-On Data Source.
   **Note:** Take note of your data source name. This information is required as you configure your database as a policy store.

6. Enter the name of the SQL Server host system in the Server field.

7. Enter the database name in the Database Name field.

8. Click Test.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The SQL Server data source is configured and appears in the System Data Sources list.

## Create a SQL Server Data Sources on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of data source entry, which is [CA Single Sign-On Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-On. The remaining attributes are specific to the driver.

Adding a MS SQL Server Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the Oracle or SQL drivers under [CA Single Sign-On Data Source].

Again, to configure a MS SQL Server data source, you must first create a system_odbc.ini file in the *policy_server_installation*/db directory. To do this, you need to rename sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini.

## Configure the SQL Server Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Enter the following under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 SQL Server Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSsqls27.so
   Description=DataDirect 7.1 SQL Server Wire Protocol
   Database=SiteMinder DataAddress=host_ip, 1433
   QuotedId=No
   AnsiNPW=No
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation, rather than a path with an environment variable.
     **Example:** export/smuser/siteminder

   - *SiteMinder Data*
     Specifies the SQL Server database instance name.

   - *host_ip*
     Specifies the IP Address of the SQL Server database.

   - **1433**
     Represents the default listening port for SQL Server.

4. If you are using Microsoft SQL Server 2008 to function as any CA Single Sign-On store, edit the [ODBC] section as follows:

   ```
   TraceFile=nete_ps_root/db/odbctrace.out
   TraceDll=nete_ps_root/odbc/lib/NStrc27.so
   InstallDir=nete_ps_root/odbc
   ```

   - *nete_ps_root*
     Specifies the explicit path to the Policy Server installation directory. This path cannot contain an environment variable.

5. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store session information.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select Session Server from the Database list.
   Data source settings become active.

3. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

4. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

5. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

6. Click Apply.
   The settings are saved.

7. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

8. Click OK.
   The Policy Server is configured to use the database as a session store.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Configure ODBC Databases as Audit Store

This section explains the following Audit Store configurations:

- Store Audit Logs in IBM DB2 (see page 345)
- Store Audit Logs in MySQL (see page 351)
- Store Audit Logs in Oracle (see page 357)
- Store Audit Logs in PostgreSQL (see page 368)
- Store Audit Logs in SQL Server (see page 370)

## Store Audit Logs in IBM DB2

**Contents**



### Before You Begin

Be sure that the table space page size (page_size) and the buffer pool page size settings for the database instance are each set to at least 16k. The default DB2 value for each setting is not sufficient for the audit log schema.

### Gather Database Information

Configuring a single IBM DB2 database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Consider the following items:

- Information that is prefixed with a W represents a Windows requirement.
- Information that is prefixed with a U represents a UNIX requirement.

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store. You can use the IBM DB2 Information Worksheet to record your values.

- **Database instance name** —Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account** —Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.
- **Administrative password** —Determine the password for the Administrative account.
- **IP address**—Determine the IP address of the database host system.
- **Tcp port**—Determine the port on which the database is listening.
- (W) **Data source name** —Determine the name that is to identify the data source.
- (U) **Policy Server root** —Determine the explicit path to where the Policy Server is installed.
- (U) **Package** —Determine the name of the package that is to process dynamic SQL.
- (U) **Package owner**—Determine the AuthID assigned to the package. The AuthID must have the authority to execute all SQLs in the package.
- (U) **Grant AuthID**—If you want to restrict execute privileges for the package, determine the AuthID that is granted execute permissions for the package.
  **Default wire protocol setting:** Public
- (U) **Isolation level**—Determine the method by which the system acquires and releases locks.
  **Default wire protocol setting:** CURSOR_STABILTY
- (U) **Dynamic sections**—Determine the number of sections that the wire protocol driver package can prepare for a single user.
  **Default wire protocol setting:** 100

## Create the Audit Store Schema

You create the CA Single Sign-On schema so that an IBM DB2 database can store audit logs.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*\db\tier2\DB2.
   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file and copy the contents to a text editor:

   ```
   sm_db2_logs.sql
   ```
4. Remove NULL from the following lines:

   ```
   sm_assertion_id         VARCHAR(255) NULL,
   sm_assertion_issuerid   VARCHAR(255) NULL,
   ```

```
sm_assertion_destinationurl     VARCHAR(4096) NULL,
sm_assertion_statuscode         VARCHAR(255) NULL,
sm_assertion_NotOnBefore    TIMESTAMP,
sm_assertion_notonorafter       TIMESTAMP,
sm_assertion_sess_starttime     TIMESTAMP,
sm_assertion_sess_notonorafter  TIMESTAMP,
sm_assertion_authcontext        VARCHAR(255) NULL,
sm_assertion_versionid          VARCHAR(255) NULL,
sm_assertion_claims             VARCHAR(255) NULL,
sm_application_name             VARCHAR(255) NULL,
sm_tenant_name                  VARCHAR(255) NULL,
sm_authentication_method        VARCHAR(255) NULL
```

5. Save the changes to the file.

6. Paste the contents into a query and execute the query.

   **Note:** For more information executing a query, see the IBM documentation.

   The audit store schema is added to the database.

## Configure an IBM DB2 Data Source for CA Single Sign-On

If you are using ODBC, configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## Create a DB2 Data Source on Windows Systems

When using ODBC, you can create a DB2 data source for the DB2 wire protocol driver.

**Follow these steps:**

1. Complete one of the following steps:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the *install_home*\Windows\SysWOW64.

     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab and click Add.

3. Scroll down and select CA Single Sign-On DB2 Wire Protocol and click Finish.

4. In the ODBC DB2 Wire Protocol Driver Setup dialog, under the General tab, complete the following steps:

a. In the Data Source Name field, enter any name.
**Example**:

```
SiteMinder DB2 Wire Data Source
```

b. (Optional) In the Description field, enter a description of the DB2 wire protocol data source.

c. In the IP Address field, enter the IP Address where the DB2 database is installed.

d. In the Tcp Port field, enter the port number where DB2 is listening on the system.

e. Click Test Connect.
The connection is tested.

5. Click OK.
The ODBC DB2 Wire Protocol Driver Setup dialog closes, the selections are saved, and the DB2 data source is created on a Windows System.

**Note:** You can now configure CA Single Sign-On to use the data source that you created.

## Create a DB2 Data Source on UNIX Systems

The CA Single Sign-on ODBC data sources are configured using a system_odbc.ini file, which you can create by renaming db2wire.ini, located in policy_server_home/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-on.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the “=” refers to a subsequent section of the file describing each individual data source. After the “=” is a comment field.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-on. The remaining attributes are specific to the driver.

Adding a DB2 Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the DB2 driver under [CA Single Sign-on Data Source].

Again, to configure a DB2 data source, you must first create a system_odbc.ini file in thepolicy_server_home/db directory. To do this, you need to rename db2wire.ini, located inpolicy_server_home/db, to system_odbc.ini.

**Note:** policy_server_home specifies the Policy Server installation path.

## Configure the DB2 Wire Protocol Driver

The following table contains configuration parameters for DB2 data sources. You can edit these parameters to configure data sources for separate key, audit log, session, and sample users databases.

| Parameter | Description | How to Edit |
|---|---|---|
| Data Source Name | Name of the data source. | Enter the data source name inside the square brackets. |
| Driver | Full path to the CA Single Sign-on DB2 Wire Protocol driver. | Replace “nete_ps_root” with the CA Single Sign-on installation directory. |
| Description | Description of the data source. | Enter any desired description. |
| Database | Name of the DB2 UDB database. | Replace “nete_database” with the name of the database configured on the DB2 UDB server. |
| LogonID | Username required for accessing the database. | Replace “uid” with the username of the DB2 UDB administrator. |
| Password | Password required for accessing the database. | Replace “pwd” with the password of the DB2 UDB administrator. |
| IPAddress | IP address or hostname of the DB2 UDB server. | Replace “nete_server_ip” with the IP address or the hostname of the DB2 UDB server. |
| TcpPort | TCP port number of the DB2 UDB server. | Replace the default value of 50000 with the actual TCP port number of the DB2 UDB server. |
| Package | The name of the package to process dynamic SQL. | Replace “nete_package” with the name of the package you want to create. |
| PackageOwner | (Optional) The AuthID assigned to the package. | Empty by default. This DB2 AuthID must have authority to execute all SQLs in the package. |
| GrantAuthid | The AuthID granted execute privileges for the package. | “PUBLIC” by default. Specify the desired AuthID if you wish to restrict the execute privileges for the package. |
| GrantExecute | Specifies whether to grant execute privileges to the AuthID listed in GrantAuthid. | Can be either 1 or 0. Set to 0 by default. |
| IsolationLevel | The method by which locks are acquired and released by the system. | CURSOR_STABILITY by default. |
| DynamicSections | The number of statements that the DB2 Wire Protocol driver package can prepare for a single user. | 100 by default. Enter the desired number of statements. |

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store audit logs.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.
2. Select ODBC from the Storage list.
   ODBC settings appear.
3. Select Audit Logs from the Database list.
4. Select ODBC from the Storage list.
   Data source settings become active.
5. Enter the name of the data source in the Data Source Information field.
   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.
   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.
6. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.
7. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

8. Click Apply.
   The settings are saved.
9. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.
10. Click OK.
    The Policy Server is configured to use the database as an audit logging database.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Store Audit Logs in MySQL

**Contents**



## Before You Begin

Be sure that MySQL is installed using the default character set (Latin1). If MySQL was not installed using the default character set, reinstall MySQL before configuring the data store.

## Gather Database Information

Gather the following information before configuring the policy store or any other type of CA Single Sign-On data store:

- **Database host**—Identify the name of the database host system.
- **Database name**—Identify the name of the database instance that is to function as the policy store or data store.
- **Database port**—Identify the port on which the database is listening.
- **Administrator account**—Identify the login ID of an administrator account with permission to manage objects in the database.
- **Administrator password** —Identify the password for the administrator account.

## Create the Audit Log Schema

You create the audit log schema so the MySQL database can store audit logs.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to the following location:

```
siteminder_home\db\tier2\MySQL.
```

- **siteminder_home**
  Specifies the Policy Server installation path.

3. Open the following file in a text editor:

   ```
   sm_mysql_logs.sql
   ```

4. Locate the following lines:

   ```
   DROP FUNCTION IF EXISTS `databaseName`.`getdate` $$
   CREATE FUNCTION `databaseName`.`getdate` () RETURNS DATE
   ```

5. Replace each instance of 'databaseName' with the name of the database functioning as the audit store.

6. Copy the contents of the entire file.

7. Paste the file contents into a query and execute the query.
   The audit store schema is created.

## Configure a MySQL Data Source for CA Single Sign-On

You configure a data source to let the Policy Server communicate with the CA Single Sign-On data store.

**Note**: If you are using MySQL 5.1.x, ensure that you assign the TRIGGER permission to the user name that is used to create the DSN.

## Create a MySQL Data Source on Windows

You create a MySQL data source for the MySQL wire protocol driver.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Do one of the following steps:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.

   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the *install_home*\Windows\SysWOW64.

     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.

3. Click System DSN.

4. Click Add.
5. Scroll down and select CA Single Sign-On MySQL Wire Protocol and click Finish.
6. Complete the following steps in the General tab:
    a. Enter a data source name in the Data Source Name field.
    **Example:**

    ```
    CA SiteMinder® MySQL Wire Data Source
    ```

    b. Enter the name of the MySQL database host system in the Host Name field.
    c. Enter the port on which the MySQL database is listening in the Port Number field.
    d. Enter the name of the MySQL database in the Database Name field.
7. Click Test Connect.
8. Click OK.
    The data source is created and appears in the System Data Sources list.

**Note:** You can now point the Policy Server to the CA Single Sign-On data store.

## Create a MySQL Data Source on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming mysqlwire.ini to system_odbc.ini. The mysqlwire.ini file is located in *siteminder_home*/db.

- **siteminder_home**
  Specifies the Policy Server installation path.

This system_odbc.ini file contains all of the names of the available ODBC data sources and the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add other data sources to this file, such as defining other ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** The value of the first line of data source entry is required when you configure the database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver that is loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

Adding a MySQL Server Data source includes:

- A new data source name in the [ODBC Data Sources] section of the file.
- A section that describes the data source using the same name as the data source.

Update the system_odbc.ini file when creating a new service name. You have entries for the MySQL driver under [CA Single Sign-On Data Source].

Again, to configure a MySQL Server data source, you create the system_odbc.ini file by renaming mysqlwire.ini to system_odbc.ini.

## Create the MySQL Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Enter the following line under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 MySQL Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSmysql27.so
   Description=DataDirect 7.1 MySQL Wire Protocol
   Database=database_nameHostName=host_nameLogonID=root_userPassword=root_user_pass
   wordPortNumber=mysql_port
   ```

   - *nete_ps_root*
     Specifies the Policy Server installation path. Enter this value as an explicit path, rather than one with an environment variable.
     **Example:** /export/smuser/siteminder
   - *database_name*
     Specifies the name of the MySQL database that is to function as the data store.
   - *host_name*
     Specifies the name of the MySQL database host system.
   - *root_user*
     Specifies the login ID of the MySQL root user.
   - *root_user_password*
     Specifies the password for the MySQL root user.
   - *mysql_port*
     Specifies the port on which the MySQL database is listening.

4. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store audit logs.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Audit Logs from the Database list.

4. Select ODBC from the Storage list.
   Data source settings become active.

5. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

6. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

7. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

8. Click Apply.
   The settings are saved.

9. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

10. Click OK.
    The Policy Server is configured to use the database as an audit logging database.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Store Audit Logs in Oracle

**Contents**



## Gather Database Information

Configuring a single Oracle database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

Information prefixed with (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. This information is required when configuring Oracle data source for UNIX.

### Required Information

Gather the following required information before configuring a supported Oracle or Oracle RAC database as a policy store or any other type of CA Single Sign-On data store:

- (U) **Policy Server installation path**—Identify the explicit path to where the Policy Server is installed.
- **Data source**—Determine the name you will use to identify the Oracle data source.
  **Example:** SM Oracle Server Wire DS.
- **Database administrative account**—Determine the user name of an account with privileges to create, read, modify, and delete objects in the database.

  **Note:** Ensure the administrative account does not have the DB role. Audit-based reports will not return correct results if the administrative account has the DB role.
- **Database administrative Password**—Determine the password for the Administrative account.

### Oracle Database Information

Gather the following information only if you are configuring a supported Oracle database as a policy store or any other type of CA Single Sign-On data store:

- **Oracle machine name**—Determine the name of the machine on which the Oracle database is installed.
- **Oracle instance service name**—Determine the service name of the database instance to which you will connect. The tnsnames.ora file specifies service names.

- **Oracle port number**—Determine the port number on which the Oracle database is listening.

## Oracle RAC Database (without SCAN) Information

Gather the following information if you are configuring a supported Oracle RAC database (without SCAN functionality configured) as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name**—Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
(Description =
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername1)(PORT=1521
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername2)(PORT=1521)
(ADDRESS = PROTOCOL = TCP)(HOST = nete_servername3)(PORT=1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA=
(SERVER = DEDICATED)
(SERVER_NAME = SMDB))
)
```

- **Oracle RAC node service names**—Determine the service names for each node in the system.
- **Oracle RAC node IP addresses**—Determine the IP Address of each node in the Oracle RAC system.

  **Note:** If you are using Oracle RAC 10g, determine the virtual IP address of each node in the system.

- **Oracle RAC node port numbers**—Determine the port number for each node in the Oracle RAC system.

## Oracle RAC Database (Using SCAN) Information

The Oracle RAC Single Client Access Name (SCAN) feature provides a single name for clients to access any Oracle Database running in a cluster.

Gather the following information if you are configuring an Oracle RAC database with SCAN functionality as a policy store or any other CA Single Sign-On data store:

- **Oracle RAC system service name**—Determine the service name for the entire system.
  **Example:** In the following tnsnames.ora file, SMDB is the service name for the entire system:

```
SMDB=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SMDB)
    )
  )
```

- **Oracle RAC SCAN Address**—Determine the FQDN of the Oracle RAC system SCAN.
- **Oracle RAC SCAN port number**—Determine the port number for the Oracle RAC system SCAN.

## Create the Audit Log Schema

You create the audit log schema so the Oracle database can store audit logs.

**To create the CA Single Sign-On schema**

1. Log into Oracle with sqlplus or some other Oracle utility as the user who administers the Policy Server database information.

   **Note:** We recommend that you do not create CA Single Sign-On schema with the SYS or SYSTEM users. If necessary, create an Oracle user, such as SMOWNER, and create the schema with that user.

2. Import the following script:
   $NETE_PS_ROOT/db/sql/sm_oracle_logs.sql
   **Note:** Environment variables may not function in Oracle's SQL utility. If you experience problems importing the script using the utility, specify an explicit path.

3. Create a table space for the audit log schema.

4. Create a user with the following privileges to manage the table space in the database:

- Connect
- Resource
- Unlimited Tablespace

## Configure an Oracle Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source for the Oracle wire protocol driver.

## Create an Oracle Data Source on Windows

Create an ODBC data source for an Oracle database.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:

     a. Navigate to the *install_home*\Windows\SysWOW64.

     b. Double–click odbcad32.exe

The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the machine name where the Oracle database is installed in the Host Name field.

6. Enter the port number where the Oracle database is listening on the machine in the Port Number field.

7. Enter the name of the Oracle instance to which you want to connect in the SID field.

   **Note:** The service name is specified in the tnsnames.ora file. The SID is the system identifier for the database instance. The tnsnames.ora file contains service names and details that Oracle uses to identify and connect to Oracle instances.

   **Example:** if the tnsnames.ora file contains the following entry for an Oracle instance, you enter instance1 in the SID field:

```
instance1 =
    (Description=
    (Address = (PROTOCOL = TCP)(Host = myhost)(Port=1521))
    (Connect_DATA_ = (SID = SIDofinstance1))
    )
```

8. Click Test Connection.
   The connection settings are tested and a prompt appears specifying that the connection is successful.

9. Click OK.
   The Oracle data source is configured for the wire protocol driver.

## Create an Oracle RAC (no SCAN) Data Source on Windows

Create an ODBC data source for an Oracle RAC database that does not use the SCAN feature.

**Follow these steps:**

1. Do one of the following:

- If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
- If you are using a supported 64–bit Windows operating system:
  a. Navigate to the C:\Windows\SysWOW64.
  b. Double–click odbcad32.exe

The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA Single Sign-On Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the IP Address of the first node in the Oracle RAC system in the Host field.
   Oracle RAC 10g: Enter the virtual IP Address.

6. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains 3 nodes:

```
SMDB=
    (Description =
(ADDRESS = (Protocol = TCP)(HOST = nete_servername1)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername2)(PORT = 1521))
(ADDRESS = (Protocol = TCP)(HOST = nete_servername3)(PORT = 1521))
(LOAD_BALANCE = yes)
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = SMDB)
)
```

7. Click the Failover tab.
   Failover settings appear.

8. Specify the host name or virtual IP Address, port number, and service name for the remaining Oracle RAC nodes in the environment in the Alternate Servers field.

   **Note:** The ServiceName is the service name for the entire Oracle RAC system.

9. Specify the AlternateServers to provide connection failover to the other Oracle nodes if the primary server is not accepting connections. The entry should have the following format: (HostName=nete_servername2:PortNumber=1521:ServiceName=nete_servicename[,...])

10. Select LoadBalancing.

11. Click OK
    The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle RAC SCAN Data Source on Windows

Create an ODBC data source for an Oracle RAC database that uses the SCAN feature.

**Follow these steps:**

1. Do one of the following:

   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to the C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe

   The ODBC Data Source Administrator appears.

2. Click the System DSN tab, and then click Add.
   The Create New Data Source dialog appears.

3. Select CA CA Single Sign-on Oracle Wire Protocol, and click Finish.
   The ODBC Oracle Wire Protocol Driver Setup dialog appears. The General tab is pulled to the front.

4. Enter a name that identifies the data source in the Data Source Name field.

   **Note:** Record this name. You will need the data source name when pointing the Policy Server to the database.

5. Enter the FQDN or IP Address of the SCAN in the Host field.

6. Enter the port number of the SCAN in the Port Number field.

7. Enter the service name for the entire Oracle RAC system in the Service Name field.
   **Example:** In the following tnsnames.ora file, the SMDB value is the service name for the entire Oracle RAC system, which contains the SCAN:

```
SMDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = clus-scan.rac.com)(PORT = 1521))
```

```
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = ORCL)
  )
)
```

8. Click OK
   The Oracle RAC data source is configured for the wire protocol driver.

## Create an Oracle Data Source on UNIX Systems

You configure the names of available ODBC data sources and the attributes that are associated with these data sources in the system_odbc.ini file.

**To create the system_odbc.ini file:**

1. Navigate to *policy_server_installation*/db
2. Rename oraclewire.ini to "system_odbc.ini".

Customize the system_odbc.ini file for each site. You can also add more data sources to this file, such as defining extra ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the “=” refers to a subsequent section of the file describing each individual data source. After the “=” is a comment field.

**Note:** If you modify of the first line of the data source entry ([CA Single Sign-On Data Source]), take note of the change. This value is required to configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when CA Single Sign-On uses this data source. The remaining attributes are specific to the driver.

**To add an Oracle Data source**:

1. Define a new data source name in the [ODBC Data Sources] section of the file.
2. Add a section that describes the data source using the same name as the data source.

To create a service name or use a different driver, edit the system_odbc.ini file. Entries for the SQL Server or Oracle drivers belong under [CA Single Sign-On Data Source].

## Configure the Oracle Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.
- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.
- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.
- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.
2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so (http://nsora27.so/)Description=DataDirect
    7.1 Oracle Wire Protocol
   LoginID=uidPassword=pwdHostName=host_namePortNumber=1521
   SID=server_idCatalogOptions=0
   ProcedureResults=0
   EnableDisableParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation.
   - *uid*
     Specifies the user name of the database account that has full access rights to the database.

- *pwd*
  Specifies the password for the database account that has full access rights to the database.

- *host_name*
  Specifies the name of the Oracle database host system.
  *server_id*
  Specifies the Oracle instance service name (SID). The SID is the system identifier for the database instance.

**Example:** In the following sample tnsnames.ora file, the value instance1 is the SID

```
instance1 =
(Description =
(ADDRESS = (Protocol = TCP)(Host = myhost)(Port = 1521)
(CONNECT_DATA = (SID = instance1))
)
```

3. Save the file.

The Oracle wire protocol driver is configured.

## Configure the Oracle Wire Protocol Driver for Oracle RAC without SCAN

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.

- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Depending on the data source you are configuring, edit the applicable data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value may cause ODBC connections to fail.

   - Add ServiceName=*nete_servicename*
   - Add AlternateServers=
   - Add Loadbalancing=1
   - Remove or comment SID=nete_serverid

   The modified text for the data source should appear as follows:

   ```
   Driver=nete_ps_root/odbc/lib/NSora27.so
   Description=DataDirect 7.1 Oracle Wire Protocol
   Logon=uidPassword=pwdHostName=server_name1PortNumber=1521
   ServiceName=service_nameCatalogOptions=0
   ProcedureRetResults=0
   EnableDescribeParam=0
   EnableStaticCursorsForLongData=0
   ApplicationUsingThreads=1
   AlternateServers=
   LoadBalancing=1
   ```

   - *nete_ps_root*
     Specifies an explicit path to the directory where Policy Server is installed.
   - *uid*
     Specifies the user name of the database account that has full access rights to the database.
   - *pwd*
     Specifies the password for the database account that has full access rights to the database.
   - *server_name1*
     Specifies the IP Address of the first Oracle RAC node.
     (Oracle 10g) Specifies the virtual IP Address of the first Oracle RAC node.
   - *service_name*
     Specifies the Oracle RAC system service name for the entire RAC system.

- **AlternateServers=**
  If the primary server is not accepting connections, specifies the connection failover to the other Oracle nodes.
  **Example:** (HostName=nete_servername2:PortNumber=1521:
  ServiceName=nete_servicename[,...])

- **LoadBalancing=1**
  Turns on client load balancing, which helps to distribute new connections to keep RAC nodes from being overwhelmed with connection requests. When enabled, the order in which primary and alternate database servers are accessed is random.

3. Save the file.
   The Oracle wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store audit logs.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Audit Logs from the Database list.

4. Select ODBC from the Storage list.
   Data source settings become active.

5. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

6. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

7. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

8. Click Apply.
   The settings are saved.

9. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

10. Click OK.
    The Policy Server is configured to use the database as an audit logging database.

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.

2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.

3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Audit Logs in PostgreSQL

**Contents**



To configure a PostgreSQL database as a standalone audit log database, complete the following procedures:

1. Gather database information.

2. Create the audit store schema.

3. Configure a PostgreSQL Server data source for CA Single Sign-On.

4. Point the Policy Server to the database.

5. Restart the Policy Server.

### Create the Audit Log Schema

You create the logging schema so the PostgreSQL Server database can store audit logs.

**To create the audit log schema**

1. Open sm_postgresql_logs.sql in a text editor and copy the contents of the entire file.

2. Start a SQL client, such as psql, and log in as the user who administers the Policy Server database.

3. Select the database instance from the database list.

4. Paste the schema from sm_postgresql_logs.sql into the query.

5. Execute the query.
   The audit log store schema is created in the database.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store audit logs.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Audit Logs from the Database list.

4. Select ODBC from the Storage list.
   Data source settings become active.

5. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

6. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

7. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

8. Click Apply.
   The settings are saved.

9. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

10. Click OK.
    The Policy Server is configured to use the database as an audit logging database.

### Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands respectively to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

## Store Audit Logs in SQL Server

**Contents**



### Gather Database Information

Configuring a single SQL Server database to function as a policy store or any other type of CA Single Sign-On data store requires specific database information.

**Note:** Information prefixed with (W) indicates that the information is only required if the Policy Server is installed on a Windows system; (U) indicates that the information is only required if the Policy Server is installed on a UNIX system. Different information is required when configuring the SQL Server data source.

- **Database instance name**
  Determine the name of the database instance that is to function as the policy store or data store.
- **Administrative account name and password**
  Determine the user name and password of an account with privileges to create, read, modify, and delete objects in the database.

- **(W) Data source name**
  Determine the name you will use to identify the data source.
  **Example:** SM SQL Server Wire DS.

- **(W) SQL Server name**
  Determine the name of the SQL Server database that contains the instance that is to function as the policy store.

- **(U) Policy Server root**
  Determine the explicit path to where the Policy Server is installed.

- **(U) IP Address**
  Determine the IP Address of the SQL Server database.

## Create the Audit Log Schema

You create the logging schema so the SQL Server database can store audit logs.

**To create the audit log schema**

1. Open sm_mssql_logs.sql in a text editor and copy the contents of the entire file.
2. Start the Query Analyzer and log in as the user who administers the Policy Server database.
3. Select the database instance from the database list.
4. Paste the schema from sm_mssql_logs.sql into the query.
5. Execute the query.
   The CA Single Sign-On audit log store schema is created in the database.

## Configure a SQL Server Data Source for CA Single Sign-On

If you are using ODBC, you need to configure a data source to let CA Single Sign-On communicate with the CA Single Sign-On data store.

## SQL Server Authentication Mode Considerations

CA Single Sign-On data sources do not support Windows authentication. Configure the CA Single Sign-On data source with the credentials of a user that is stored in the database.

**Note:** For more information about SQL Server authentication modes, see the vendor–specific documentation.

## Create a SQL Server Data Source on Windows

ODBC requires that you configure a data source for the SQL Server wire protocol.

**Note:** This procedure only applies if the Policy Server is installed on a Windows System.

**Follow these steps:**

1. Complete one of the following steps:
   - If you are using a supported 32–bit Windows operating system, click Start and select Programs, Administrative Tools, ODBC Data Sources.
   - If you are using a supported 64–bit Windows operating system:
     a. Navigate to C:\Windows\SysWOW64.
     b. Double–click odbcad32.exe.

   The ODBC Data Source Administrator appears.
2. Click the System DSN tab.
   System data source settings appear.
3. Click Add.
   The Create New Data Source dialog appears.
4. Select CA Single Sign-On SQL Server Wire Protocol and click Finish.
   The ODBC SQL Server Wire Protocol Driver Setup dialog appears.
5. Enter the data source name in the Data Source Name field.
   **Example:** CA Single Sign-On Data Source.
   **Note:** Take note of your data source name. This information is required as you configure your database as a policy store.
6. Enter the name of the SQL Server host system in the Server field.
7. Enter the database name in the Database Name field.
8. Click Test.
   The connection settings are tested and a prompt appears specifying that the connection is successful.
9. Click OK.
   The SQL Server data source is configured and appears in the System Data Sources list.

## Create a SQL Server Data Sources on UNIX Systems

The CA Single Sign-On ODBC data sources are configured using a system_odbc.ini file, which you create by renaming sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini. This system_odbc.ini file contains all of the names of the available ODBC data sources as well as the attributes that are associated with these data sources. This file must be customized to work for each site. Also, you can add additional data sources to this file, such as defining additional ODBC user directories for CA Single Sign-On.

The first section of the system_odbc.ini file, [ODBC Data Sources], contains a list of all of the currently available data sources. The name before the "=" refers to a subsequent section of the file describing each individual data source. After the "=" is a comment field.

**Note:** If you modify of the first line of data source entry, which is [CA Single Sign-On Data Source], take note of the change because you will need this value when configure your ODBC database as a policy store.

Each data source has a section in the system_odbc.ini file describing its attributes. The first attribute is the ODBC driver to be loaded when this data source is used by CA Single Sign-On. The remaining attributes are specific to the driver.

Adding a MS SQL Server Data source involves adding a new data source name in the [ODBC Data Sources] section of the file, and adding a section that describes the data source using the same name as the data source. You need to change the system_odbc.ini file if you create a new service name or want to use a different driver. You should have entries for the Oracle or SQL drivers under [CA Single Sign-On Data Source].

Again, to configure a MS SQL Server data source, you must first create a system_odbc.ini file in the *policy_server_installation*/db directory. To do this, you need to rename sqlserverwire.ini, located in *policy_server_installation*/db, to system_odbc.ini.

## Configure the SQL Server Wire Protocol Driver

You configure the wire protocol driver to specify the settings the Policy Server uses to connect to the database.

This procedure only applies if the Policy Server is installed on a UNIX system. If you have not already done so, copy one of the following files and rename it **system_odbc.ini**:

- sqlserverwire.ini
- oraclewire.ini
- mysqlwire.ini
- postgresqlwire.ini

These files are located in *siteminder_home*/db.

The system_odbc.ini file contains the following sections. The data source that you are configuring determine the section or sections that you edit:

- **[SiteMinder Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the policy store.
- **[SiteMinder Logs Data Source]**
  Specifies the settings CA Single Sign-On is to use to connect to the database functioning as the audit log database.

- **[SiteMinder Keys Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the key store.

- **[SiteMinder Session Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the session store.

- **[SmSampleUsers Data Source]**
  Specifies the settings CA Single Sign-On is to connect to the database functioning as the sample user data store.

**Follow these steps:**

1. Open the system_odbc.ini file.

2. Enter the following under [ODBC Data Sources]:

   ```
   SiteMinder Data Source=DataDirect 7.1 SQL Server Wire Protocol
   ```

3. Depending on the data source you are configuring, edit one or more of the data source sections with the following information. When editing data source information, do not use the pound sign (#). Entering a pound sign comments the information, which truncates the value. The truncated value can cause ODBC connections to fail.

   ```
   Driver=nete_ps_root/odbc/lib/NSsqls27.so
   Description=DataDirect 7.1 SQL Server Wire Protocol
   Database=SiteMinder DataAddress=host_ip, 1433
   QuotedId=No
   AnsiNPW=No
   ```

   - *nete_ps_root*
     Specifies the explicit path of the Policy Server installation, rather than a path with an environment variable.
     **Example:** export/smuser/siteminder

   - *SiteMinder Data*
     Specifies the SQL Server database instance name.

   - *host_ip*
     Specifies the IP Address of the SQL Server database.

   - **1433**
     Represents the default listening port for SQL Server.

4. If you are using Microsoft SQL Server 2008 to function as any CA Single Sign-On store, edit the [ODBC] section as follows:

   ```
   TraceFile=nete_ps_root/db/odbctrace.out
   TraceDll=nete_ps_root/odbc/lib/NStrc27.so
   InstallDir=nete_ps_root/odbc
   ```

   - *nete_ps_root*
     Specifies the explicit path to the Policy Server installation directory. This path cannot contain an environment variable.

5. Save the file.
   The wire protocol driver is configured.

## Point the Policy Server to the Database

You point the Policy Server to the database so the Policy Server can read and store audit logs.

**To point the Policy Server to the data store**

1. Open the Policy Server Management Console, and click the Data tab.
   Database settings appear.

2. Select ODBC from the Storage list.
   ODBC settings appear.

3. Select Audit Logs from the Database list.

4. Select ODBC from the Storage list.
   Data source settings become active.

5. Enter the name of the data source in the Data Source Information field.

   - (Windows) this entry must match the name you entered in the Data Source Name field when you created the data source.

   - (UNIX) this entry must match the first line of the data source entry in the system_odbc.ini file. By default, the first line in the file is [CA Single Sign-On Data Sources]. If you modified the first entry, be sure that you enter the correct value.

6. Enter and confirm the user name and password of the database account that has full access rights to the database instance in the respective fields.

7. Specify the maximum number of database connections allocated to CA Single Sign-On.

   **Note:** We recommend retaining the default for best performance.

8. Click Apply.
   The settings are saved.

9. Click Test Connection.
   CA Single Sign-on returns a confirmation that the Policy Server can access the data store.

10. Click OK.
    The Policy Server is configured to use the database as an audit logging database.

## Restart the Policy Server

You restart the Policy Server for certain settings to take effect.

**Follow these steps:**

1. Open the Policy Server Management Console.
2. Click the Status tab, and click Stop in the Policy Server group box.
   The Policy Server stops as indicated by the red stoplight.
3. Click Start.
   The Policy Server starts as indicated by the green stoplight.

   **Note**: On UNIX, execute the stop-ps and start-ps commands to restart Policy Server. To restart Policy Server and CA Risk Authentication, execute the stop-all and start-all commands.

# Sample User Directories

**Contents**



CA Single Sign-On does not require the use of a proprietary user store. However, CA Single Sign-On does provide schema files that populate a relational database with sample users.

## Configure an IBM DB2 Sample User Directory

You configure a sample user directory to populate a database with sample users.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to *siteminder_home*\db\tier2\DB2.

   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file and copy the contents to a text editor:
   smsampleusers_db2.sql
4. Paste the contents into a query and execute the query.

   **Note:** For more information about executing a query, see the IBM documentation.

   The user directory is populated with the sample users.
5. Configure the user directory connection to the Policy Server.

## Configure a MySQL Sample User Directory

You configure a sample user directory to populate a database with sample users.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Navigate to the following location:

   ```
   siteminder_home\db\tier2\MySQL.
   ```

   - **siteminder_home**
     Specifies the Policy Server installation path.
3. Open the following file in a text editor:

   ```
   smsampleusers_mysql.sql
   ```
4. Locate the following lines:

   ```
   DROP FUNCTION IF EXISTS `databaseName`.`getdate` $$
   CREATE FUNCTION `databaseName`.`getdate` () RETURNS DATE
   ```
5. Replace each instance of 'databaseName' with the name of the database functioning as the sample user store.
6. Copy the contents of the entire file.
7. Paste the file contents into a query and execute the query.

   **Note:** For more information about executing a query, see the MySQL documentation.

   The user store is populated with the sample users.
8. Configure the user directory connection to the Policy Server.

## Configure an Oracle Sample User Directory

You configure a sample user directory to populate a database with sample users.

**To configure the sample user directory**

1. Log into Oracle with sqlplus or some other Oracle utility as the user who administers the Policy Server database information.

> **Note:** We recommend that you do not create CA Single Sign-On schema with the SYS or SYSTEM users. If necessary, create an Oracle user, such as SMOWNER, and create the schema with that user.

2. Import the following script:
   $NETE_PS_ROOT/db/sql/smsampleusers_oracle.sql
   **Note:** Environment variables may not function in Oracle's SQL utility. If you experience problems importing the script using the utility, specify an explicit path.
   The user directory is populated with the sample users.

3. Create a table space for the user directory.

4. Create a user with the following privileges to manage the table space in the database: Connect, Resource, and Unlimited Tablespace.

5. Configure the user directory connection to the Policy Server.

## Configure a SQL Server Sample User Directory

You configure a sample user directory to populate a database with sample users.

Remember to set up a SQL Server data source connection (see page 242) for the user directory.

**To configure the sample user directory**

1. Open smsampleusers_sqlserver.sql in a text editor and copy the contents of the entire file.

2. Start the Query Analyzer and log in as the user who administers the Policy Server database.

3. Select the database instance from the database list.

4. Paste the schema from smsampleusers_sqlserver.sql into the query.

5. Execute the query.
   The user directory is populated with the sample users.

6. Configure the user directory connection to the Policy Server.

# Modified Environment Variables

The Policy Server installation adds and modifies the following environment variables:

# Policy Server

- **ARCOT_HOME**
  Specifies the path to the Advanced Authentication component required for Session Assurance functionality.

- **CA_SM_PS_FIPS140**
  Specifies the FIPS mode.

- **NETE_PS_ROOT**
  Specifies the path to the Policy Server install directory.

- **NETE_PS_PATH**
  Specifies the path to the Policy Server bin, thirdparty, and lib directories.

- **NETEGRITY_LICENSE_FILE**
  Specifies the path to the license file.

- **NETE_JVM_OPTION_FILE**
  Specifies the path to the JVMOptions.txt file.

- **NETE_PS_OPACK**
  Specifies whether the Policy Server option pack is installed.

- **NETE_JRE_ROOT**
  (Windows) Specifies the registry key that specifies the Java Runtime Environment (JRE).
  (UNIX) Specifies the path to the JRE install directory.

- **NETE_JAVA_PATH**
  Specifies the path to the JRE install directory.

- **(Windows only) NETE_SHORTCUTS**
  Specifies the path to to the Windows Start menu shortcuts.

- **(WIndows only) NSPR_NATIVE_THREADS_ONLY**
  When set (with a value of 1, the default), disables a low-level threading option that is enabled by default in one of the LDAP component libraries. This option must be set to prevent intermittent Policy Server process hang or crash failures. For more information, see the related Knowledge Base article (http://www.ca.com/us/support/ca-support-online/product-content/knowledgebase-articles/tec441875.aspx).

# Administrative UI

The Administrative UI installation does not add environment variables.

# Report Server

- **IAM_RPTSRV_HOME**
  Specifies the path to the Report Server install directory.

- **BOE_SSL_JVMOPTIONS**
  Specifies Report Server Java options for SSL.

# Uninstall Policy Server

Shut down all instances of the Policy Server Management Console and complete the following steps to uninstall Policy Server:

- Uninstall Policy Server on Windows (see page 380)
- Uninstall Policy Server on UNIX (see page 382)

# Uninstall Policy Server on Windows

## Remove Policy Server References from Agent Host Files

Remove the Policy Server reference from the SmHost.conf file to prevent unexpected results from the Web Agent or CA Access Gateway once Policy Server is uninstalled.

**Follow these steps**:

1. Navigate to *web_agent_home*/config.
   ***web_agent_home***
   Specifies the installation directory of the Web Agent.

2. Open the SmHost.conf file in a text editor.

3. Delete the line that begins with "policyserver=".
   **Note**: This line contains the IP address and port numbers for the Policy Server you are uninstalling.

4. Save SmHost.conf.

5. Navigate to *sps_home*\proxy-engine\conf\defaultagent.
   ***sps_home***
   Specifies the installation directory of CA SPS.

6. Repeat **Steps 2-4**.

## Stop All CA Single Sign-On Processes

Stop all the CA Single Sign-On processes so that the Policy Server files are safely removed.

**Follow these steps:**

1. Log in to the Windows system.
2. From the Administrative Tools, open the Services.
3. Scroll down to the CA Single Sign-On Policy Server service and select Stop.

## Uninstall Policy Server

You uninstall the Policy Server when it is no longer required on the system.

**Follow these steps:**

1. Open the Windows Control Panel and go to the list of programs.
2. Right–click CA Single Sign-On Policy Server.
3. Click Uninstall/Change.
4. Follow the instructions of the wizard.

   If you are prompted to remove a shared file, click **No to All**.
5. If prompted, restart the system.

## Remove Directories, Registry Entries, and Services

Manually remove the following folders, files, registry settings, and virtual directories after uninstalling the Policy Server:

1. **Windows system**
   - *siteminder_home*\bin
   - *siteminder_home*\install_config_info
   - C:\Program Files\ZeroG Registry\com.zerog.registry.xml

   **Important!** Remove all items before reinstalling the Policy Server.
2. **AdventNet software registry entry**—Delete the AdventNet software registry entry only if the software was not on the system before installing the Policy Server. This registry entry is located in HKEY_LOCAL_MACHINE\SOFTWARE\Advent,Inc.

Manually remove the following services:

- SiteMinder Health Monitor Service

- SiteMinder Policy Server
- SNMP Agent

**Follow these steps:**

1. Stop each service.
2. Remove the following Windows registry keys:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SMServMon
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SMPolicySrv
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Agent Service
```

Policy Server is uninstalled.

# Uninstall Policy Server on UNIX

## Remove Policy Server References from Agent Host Files

Remove the Policy Server reference from the SmHost.conf file to prevent unexpected results from the Web Agent or CA Access Gateway once Policy Server is uninstalled.

**Follow these steps**:

1. Navigate to *web_agent_home*/config.
   ***web_agent_home***
   Specifies the installation directory of the Web Agent.
2. Open the SmHost.conf file in a text editor.
3. Delete the line that begins with "policyserver=".
   **Note**: This line contains the IP address and port numbers for the Policy Server you are uninstalling.
4. Save SmHost.conf.
5. Navigate to *sps_home*\proxy-engine\conf\defaultagent.
   ***sps_home***
   Specifies the installation directory of CA SPS.
6. Repeat Steps 2-4.

## Set the JRE in the PATH Variable

Set the JRE in the PATH variable when you uninstall the Policy Server, Web Agent, CA Access Gateway, SDK, or documentation to prevent the uninstallation program from stopping and issuing error messages.

**Follow these steps**:

1. Run the following command:

```
PATH=$PATH:<JRE>/bin
```

**JRE**
Specifies the location of the JRE.

2. Run the following command:

```
export PATH
```

The JRE is set in the PATH variable.

## Stop All CA Single Sign-On Processes

Stop all CA Single Sign-On processes to ensure that Policy Server files are safely removed. Use the same user account that was used to install Policy Server.

**Follow these steps**:

1. Log into the UNIX system.
2. Navigate to the *sso_home*/siteminder/directory.
3. Execute the stop-all command.

## Uninstall Policy Server

Do not manually remove the installation directories to uninstall Policy Server. Use only the uninstall shell script. If you remove only the installation directories, related registries can be left behind. If you try to re–install this component on this host system, the entries can prevent a successful installation.

**Follow these steps:**

1. Log in to the Policy Server host system as the user who installed the Policy Server.
   **Note:** The user who installed the Policy Server should have the required CA Single Sign-On scripts sourced. If the CA Single Sign-On scripts are not sourced at login, or you logged in as another user, source the following scripts:

   ```
   smprofile.ksh
   ca_ps_env.ksh
   ```

2. Change to the following directory in a console window:

   ```
   siteminder_home/siteminder/install_config_info/ca-ps-uninstall
   ```

   - **siteminder_home**
     Specifies the Policy Server installation path.

3. Run the following command:

   ```
   ./uninstall
   ```

   The uninstallation program is displayed.

4. Press Enter.
   A status indicator displays the progress.

5. Change the directory to the one above the CA Single Sign-On installation directory.
   **Example:** If the CA Single Sign-On installation directory is /export/smuser/ca/siteminder, navigate to:

   ```
   /export/smuser/ca
   ```

6. Execute the following command:

   ```
   $ rm -rf siteminder
   ```

   The CA Single Sign-On installation directory is removed.

7. Open the following file from the HOME directory:

   ```
   .profile
   ```

8. Locate and delete the line that contains smprofile.ksh.
   **Example:**

   ```
   ./export/smuser/siteminder/smprofile.ksh
   ```

9. Save the file.

## Remove CA Single Sign-On References from iPlanet Web Server (IWS)

You manually remove CA Single Sign-On references from IWS after uninstalling the Policy Server. CA Single Sign-On references are left in the obj.conf file and the magnus.conf file.

**Follow these steps:**

1. Log into an account that has privileges to access and modify the Web server's configuration.

2. Go to the following location from UNIX command line.

   ```
   <SunJavaSystem_home>/https-<hostname>/config
   ```

   The obj.conf and magnus.conf files appear in the config folder.

3. Open obj.conf and remove the following lines:

   ```
   NameTrans fn="assign-name" from="/servlet/*" name="<ServletExec_instance name>"
   NameTrans fn="assign-name" from="*.jsp*" name="<ServletExec_instance name>"
   NameTrans fn="pfx2dir" from="/sitemindermonitor" dir="/<siteminder_installation>
   /monitor"
   NameTrans fn="pfx2dir" from="/sitemindercgi" dir="/<siteminder_installation>
   /admin" name="cgi"
   NameTrans fn="pfx2dir" from="/siteminder" dir="/<siteminder_installation>/admin"
   NameTrans fn="pfx2dir" from="/netegrity_docs" dir="/netegrity
   /netegrity_documents"
   <Object name="<ServletExec_instance name>">
   Service fn="ServletExecService" group="<ServletExec_instance name>"
   </Object>
   ```

4. Save and close the obj.conf file.

5. Open magnus.conf and remove the following lines:

   ```
   Init fn="init-cgi" SM_ADM_UDP_PORT="44444" SM_ADM_TCP_PORT="44444"
   Init fn="load-modules" shlib="/<Servlet_Exec_Install>/bin/ServletExec_Adapter.
   ```

```
so" funcs="ServletExecInit,ServletExecService"
Init fn="ServletExecInit" <ServletExec_instance name>.instances="<IP_Address>:
<port_number>"
```

6. Save and close magnus.conf.

7. Restart the Web server.
   CA Single Sign-On references are removed from IWS.
   The CA Single Sign-On references no longer appear in IWS.

## Remove CA Single Sign-On References from StartServletExec

**Follow these steps:**

1. Log into the UNIX system with an account that has privileges to access and modify the configuration of ServletExec.

2. Go to the /usr/NewAtlanta/ServletExecAS/*ServletExec_instance name* folder.

3. Remove the following lines from the StartServletExecscript:

```
CLASSPATH=${NA_LIB}/servlet-api.jar:${NA_LIB}/jsp-
api.jar:${NA_LIB}/ServletExec60.jar:${NA_LIB}/ServletExecAdmin.jar:${NA_LIB}/el-
api.jar:${NA_LIB}/jasper-el.jar:${JL}/tools.jar:${NA_LIB}/jstl.jar:${NA_LIB}
/appserv-
jstl.jar:${NA_LIB}/activation.jar:${NA_LIB}/mail.jar:${HOMEDIRPATH}/classes:
/siteminder_home/monitor/
smmonui.jar:/siteminder_home/lib/smconapi.jar:/siteminder_home/lib
/smmonclientapi.jar
$SENAME $HOMEDIR $MIMEFILE $DOCROOTDIR -allow 127.0.0.1 -port $PORT $SEOPTS"
$SENAME $HOMEDIR $MIMEFILE $DOCROOTDIR -allow 127.0.0.1 -port $PORT $SEOPTS -
addl
"/sitemindermonitor=/siteminder_home/monitor""
```

   - **siteminder_home**
     Specifies the Policy Server installation path.

4. Save and close the StartServletExecscript.

5. Restart ServletExec.
   The uninstallation is complete.

## Remove System Registry Entries

Remove the CA Single Sign-On entries in the com.zerog.registry.xml file. You can locate the file in one of the following folders:

- $HOME/.com.zerog.registry.xml
- /var/.com.zerog.registry.xml

Policy Server is uninstalled.

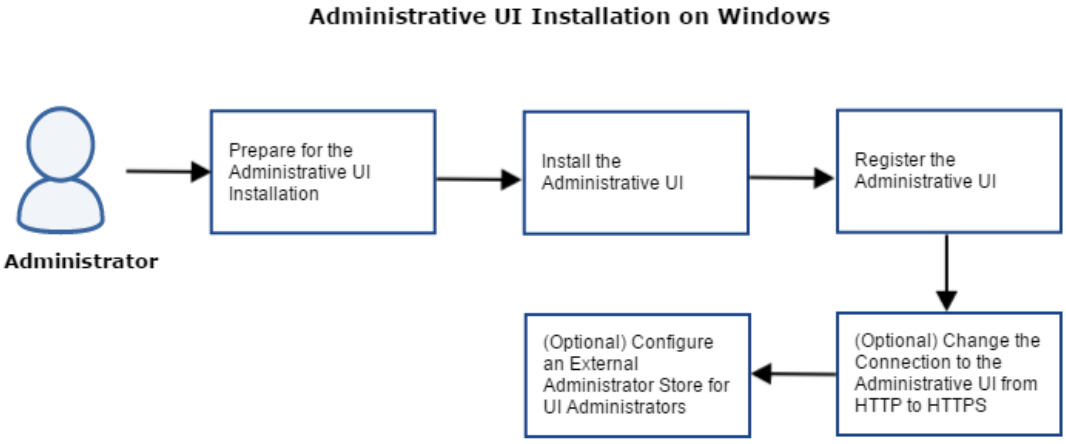
# Install Administrative User Interface

An application server is required to run the Administrative User Interface (Administrative UI). Use one of the following options to install Administrative UI:

- **Stand-alone installation**—The installation is of two steps:
  1. Run the prerequisites installer that installs an embedded JBoss application server and the required JDK.
  2. Run the Administrative UI installer.
- **External application server installation**—The installation is of two steps:
  1. Install one of the following application servers:
     - JBoss Application Server
     - WebLogic Server
     - Websphere Application Server
  2. Run the Administrative UI installer.

# Install the Administrative UI on Windows (stand-alone)

The following diagram describes how to complete a stand-alone installation of Administrative UI on Windows:




Complete the following steps:

# Prepare for the Administrative UI Installation

Complete the following tasks to prepare your system for installing Administrative UI.

## Verify the Windows UI Host System Requirements

A Windows host system for a stand-alone Administrative UI installation must meet the following minimum system requirements:

- **CPU:** x86 or x64, 1.2 GHz or better.
- **Memory:** 1 GB of system RAM. We recommend 2 GB.
- **Available disk space:** 840 MB.
- **Temp directory space:** 3 GB.
- **Screen resolution:** 1024 x 768 or higher resolution with 256 colors or better to view the Administrative UI properly.

Use the Platform Support Matrix to verify that the operating environment and other required third-party components are supported.

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Gather Information for the Installer

Gather the following information before installing and registering the Administrative UI:

- **Installation location**—Determine the Administrative UI installation path.
- **Administrative UI system name**—Identify the fully qualified name of the Administrative UI host system.
- **Server port**—Identify the port on which JBoss must listen for HTTP requests.

- **Super user account password**—Identify the password for the default user account (siteminder).
- **Policy Server system name**—Identify the following:
  - The Policy Server to which the Administrative UI will be registered.
  - The fully qualified name of the Policy Server host system.
- **Policy Server authentication port**—If you changed the default settings after installing the Policy Server, identify the Policy Server authentication port. The Settings tab in the Policy Server Management Console lists the access control ports.

## Reset the Administrative UI Registration Window

Reset the Administrative UI registration window if you are installing Administrative UI after 24 hours of performing *one* of the following steps:

- Configured a policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder_administrator[:passphrase] -adminui-setup -t timeout -r
   retries -c comment -cp
   -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   ***siteminder_administrator***

   Defines the administrator.

   ***passphrase***

   Defines the password for the administrator account.

   ***-adminui-setup***

   Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   ***-t timeout***

   (Optional) Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.

   Default: 1440 (24 hours)

   Minimum: 1

   Maximum: 1440 (24 hours)

**-r retries**

(Optional) Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.

Default: 1

Maximum: 5

**-c comment**

(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-cp**

(Optional) Specifies that the registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-l log_path**

(Optional) Specifies where the registration log file must be exported.

Default: siteminder_home\log

***-e error_path***

(Optional) Sends exceptions to the specified path.

Default: stderr

**-vT**

(Optional) Sets the verbosity level to TRACE.

**-vI**

(Optional) Sets the verbosity level to INFO.

**-vW**

(Optional) Sets the verbosity level to WARNING.

**-vE**

(Optional) Sets the verbosity level to ERROR.

**-vF**

(Optional) Sets the verbosity level to FATAL.

XPSRegClient supplies the administrator credentials to Policy Server. Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Review the Installation Prerequisites

Consider the following items before you install the Administrative UI:

- Install the Administrative UI using the installation media on the Technical Support site. For a list of installation media names, see Release Notes.
- Run the installer from the Administrative UI host system. Do not run the installer from a mapped network share or UNC path.
- Extract the executable of the prerequisite installer and the Administrative UI installer to the same location.
- The installation zip contains a layout.properties file. If you move the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

# Install the Administrative UI

Run the prerequisite installer followed by the Administrative UI installer.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the prerequisite installation media.
3. Ensure that you have the local administrator privileges to run the installer. Double-click *prerequisite_installation_media*.
   *prerequisite_installation_media* specifies the prerequisite installer executable for the Administrative UI.
   The installer starts.
4. Click Install.
   The required components are installed.
5. Click Done.
   The Administrative UI installer starts.
6. Follow the prompts and click Install.
   The Administrative UI is installed. After the installation is complete, the Administrative UI starts automatically and the login screen displays.

# Register the Administrative UI

The Administrative UI is registered with Policy Server when you log in to it for the first time with the default super user account (siteminder) credentials. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. Policy Server is managed using the Administrative UI according to the administrative privileges of the user.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configured the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to Policy Server. Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

Review the following considerations before you start the Administrative UI for the first time:

- The first time that you launch an Administrative UI over SSL, the browser warns that a trusted company did not issue the security certificate. This warning relates to a self-signed certificate that is generated during SSL registration. Approve the certificate and proceed.
- The Administrative UI requires that you enable JavaScript in the browser. If you use IE 11 to access the Administrative UI, you might see a message that the website content is blocked. From this message, add the Administrative UI as a trusted site, where JavaScript is enabled by default. If you clear the check box associated with the message you can log in to the UI, but it does not render correctly unless you enable JavaScript. Enable JavaScript for the security zone that the UI is in or add the UI as a trusted site. To add a trusted site, begin at the IE menu and select Tools, Internet Options. From the Security tab, select Trusted Sites and add the UI.
- When using the task pane on the right, always save your changes before opening or closing the menu pane on the left or navigating elsewhere.
- Do not use the Refresh or Back buttons of the browser while using the Administrative UI. Using these buttons resubmits the form, and creates an invalid state.

**Follow these steps:**

1. Open a web browser.
   If the Administrative UI is installed on a Windows system, you can start the Administrative UI on that system by clicking the SSO Administrative Console shortcut in the CA program group.
2. Enter the location of the Administrative UI using the following guidelines:
   - If the Administrative UI was installed using the standalone option and the Administrative UI was registered over SSL, use the following URL format:

     `https://host.domain:8443/iam/siteminder/adminui`
   - If the Administrative UI was installed using the standalone option and the Administrative UI was not registered over SSL, use the following URL format:

     `http://host.domain:8080/iam/siteminder/adminui`

- If the Administrative UI was installed to an existing application server infrastructure and the Administrative UI was registered over SSL, use the following URL format:

  `http://host.domain:port/iam/siteminder/adminui`

- If the Administrative UI was installed to an existing application server infrastructure and the Administrative UI was not registered over SSL, use the following URL format:

  `https://host.domain:port/iam/siteminder/adminui`

  ***host*** specifies the name of the Administrative UI host system.
  ***domain*** specifies the fully qualified domain name of the Administrative UI host system.
  ***port*** specifies the port on which the application server listens for requests.

3. In the login screen, enter **siteminder** in **User Name**.
4. Enter the siteminder account password in Password.

   **Note**: If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password**.

5. Enter the fully qualified Policy Server host name in Server.
   Consider the following points:

   - You can enter a valid IPv4 address or IPv6 address.
   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

Configure additional Policy Server connections for the Administrative UI, or proceed to install an agent.

If you encountered any installation issues, use the following log files to troubleshoot the issues:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  **Default Location**: *administrative_ui_home*\adminui\install_config_info
- CA_ SiteMinder_Administrative_UI_InstallLog.log
  **Default Location**: *administrative_ui_home*\adminui\install_config_info

# (Optional) Change the Connection to the Administrative UI from HTTP to HTTPS

If you used HTTP to register the Administrative UI with Policy Server, you can change the connection type from HTTP to HTTPS. Modify the context.xml file of the embedded JBoss application server to enable secure cookies.

**Follow these steps:**

1. Shut down the application server.
2. Navigate to the following location: user_console.war\WEB-INF
3. Open the **context.xml** file.
4. Add the secure="true" attribute to the <SessionCookie> tag. For example:

   ```
   <SessionCookie secure="true" httpOnly="true" />
   ```

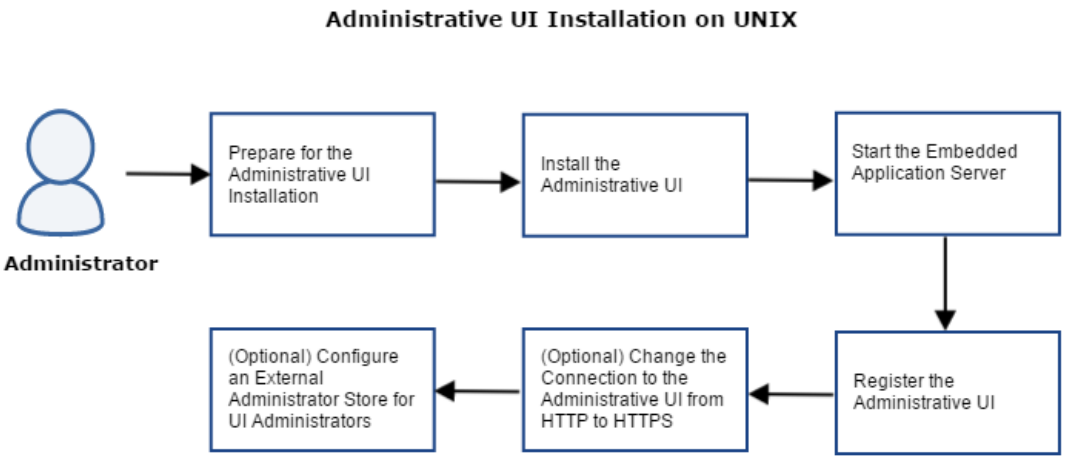
5. Save and close the file.
6. Restart the application server.

## (Optional) Configure an External Administrator Store for UI Administrators

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# Install the Administrative UI on UNIX (stand-alone)

The following diagram describes how to complete a stand-alone installation of Administrative UI on UNIX:




Complete the following steps:

- Prepare for the Administrative UI Installation (see page 394)
  - Verify the UNIX UI Host System Requirements (see page 394)

# Prepare for the Administrative UI Installation

Complete the following tasks to prepare your system for installing Administrative UI.

## Verify the UNIX UI Host System Requirements

A UNIX Administrative UI host system for the stand–alone installation must meet the following minimum system requirements::

- **CPU**
  - Solaris: UltraSparc, 440 MHz or better.
  - Red Hat Linux: x86 or x64, 700 MHz or better.
    The Red Hat 6 operating system relies on entropy for performance. Before you install the component, increase entropy. Otherwise, the installation takes an exceedingly long time, and the Administrative UI responses slow down during runtime. We recommend using the following command:

    ```
    mv /dev/random /dev/random.org
    ln -s /dev/urandom /dev/random
    ```

- **Memory:** 1 GB of system RAM. We recommend 2 GB.
- **Available disk space:** 840 MB.
- **Temp directory space:** 3 GB.
- **Screen resolution:** 1024 x 768 or higher resolution with 256 colors or better to view the Administrative UI properly.

Use the Platform Support Matrix to verify that the operating environment and other required third-party components are supported.

## Increase Entropy

By default, Red Hat uses entropy that is obtained from general computing operations, which generate random numbers. The random numbers that the Red Hat default random number generator generates are available using the following character devices:

- /dev/random. This is the most secure device as it stops supplying numbers when the entropy amount is insufficient for generating a good random output.
- /dev/urandom. This reuses the entropy pool of the kernel and supplies unlimited pseudo-random numbers with less entropy.

Policy Server uses the /dev/random character device for key generation. However, as /dev/random stops supplying numbers when entropy is insufficient, it might impact the Policy Server run time performance.

To increase the source of randomness for the entropy pool, use one of the following options:

- Most secure and FIPS compliant: Install a hardware entropy generator and configure the rngd daemon to use it to populate /dev/random.
  Example: rngd -r /dev/*device_name* -o /dev/random -b
  ***device_name*** is the character device in use. The device name varies depending on the hardware random number generator that you are using, for example, /dev/hwrng.
  For information about the rngd daemon, see the Red Hat documentation.
- Good security and not FIPS compliant: Configure the rngd daemon to populate /dev/random. Execute the following command:

  ```
  rngd -r /dev/urandom -o /dev/random -b
  ```

  Third-party alternatives to the rngd entropy daemon are also available.
- Least secure and not FIPS compliant: Configure a symbolic link between /dev/urandom and /dev /random. Execute the following commands:

  ```
  mv /dev/random /dev/random.org (http://random.org)
  ln -s /dev/urandom /dev/random
  ```

  **Important!** To ensure that sufficient entropy is available for Policy Server after a system crash or reboot, add your chosen option to an appropriate startup or service script.

To monitor the entropy on the system, execute the following command:

```
watch -n 1 cat /proc/sys/kernel/random/entropy_avail
```

## Required Linux Libraries

CA Single Sign-On requires certain Linux libraries for components that operate on Linux. We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

The following list describes the commands to install the required libraries on the host system:

**Red Hat 5.x**

```
yum install -y compat-gcc-34-c++
yum install -y libidn.i686
yum install -y libstdc++.i686
yum install -y ncurses-libs.i686
```

**Red Hat 6.x**

```
yum install -y libstdc++.i686
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y ncurses-libs.i686
yum install -y libXrender.i686
yum install -y libXtst.i686
```

**Additional Packages for Red Hat 6.x 64-bit**

```
yum install -y libXau.i686
yum install -y libXext.i686
yum install -y libxcb.i686
yum install -y compat-libstdc++-33.i686
yum install -y compat-db42.i686
yum install -y compat-db.i686
yum install -y compat-db43.i686
yum install -y libXi.i686
yum install -y libX11.i686
yum install -y libXtst.i686
yum install -y libXrender.i686
yum install -y libXft.i686
yum install -y libXt.i686
yum install -y libXp.i686
yum install -y libstdc++.i686
yum install -y libICE.i686
yum install -y compat-libtermcap.i686
yum install -y libidn.i686
yum install -y libSM.i686
yum install -y libuuid.i686
```

If the correct library is unavailable, CA Single Sign-On displays the following error:

```
java.lang.UnsatisfiedLinkError
```

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Gather Information for the Installer

Gather the following information before installing and registering the Administrative UI:

- **Installation location**—Determine the Administrative UI installation path.
- **Administrative UI system name**—Identify the fully qualified name of the Administrative UI host system.
- **Server port**—Identify the port on which JBoss must listen for HTTP requests.
- **Super user account password**—Identify the password for the default user account (siteminder).
- **Policy Server system name**—Identify the following:

- The Policy Server to which the Administrative UI will be registered.
- The fully qualified name of the Policy Server host system.

- **Policy Server authentication port**—If you changed the default settings after installing the Policy Server, identify the Policy Server authentication port. The Settings tab in the Policy Server Management Console lists the access control ports.

## Reset the Administrative UI Registration Period on UNIX

Reset the Administrative UI registration window if you are installing Administrative UI after 24 hours of performing *one* of the following steps:

- Configured a policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

**Follow these steps**:

1. If the environment variables are not set, complete the following procedure:
   a. Log in to the Policy Server host system.
   b. Open a shell and navigate to *siteminder_home*.
   c. Run the following command:

      ```
      smprofile.ksh
      ```

2. Log in to the Policy Server host system.
3. Run the following command:

   ```
   XPSRegClient siteminder_administrator[:passphrase] -adminui-setup -t timeout -
   r retries -c comment -cp
   -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   ***siteminder_administrator***

   Defines the administrator.

   ***passphrase***

   Defines the password for the administrator account.

   ***-adminui-setup***

   Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   ***-t timeout***

   (Optional) Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.

Default: 1440 (24 hours)

Minimum: 1

Maximum: 1440 (24 hours)

**-r retries**

(Optional) Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.

Default: 1

Maximum: 5

**-c comment**

(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-cp**

(Optional) Specifies that the registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-l log_path**

(Optional) Specifies where the registration log file must be exported.

Default: siteminder_home\log

***-e error_path***

(Optional) Sends exceptions to the specified path.

Default: stderr

**-vT**

(Optional) Sets the verbosity level to TRACE.

**-vI**

(Optional) Sets the verbosity level to INFO.

**-vW**

(Optional) Sets the verbosity level to WARNING.

**-vE**

(Optional) Sets the verbosity level to ERROR.

**-vF**

(Optional) Sets the verbosity level to FATAL.

XPSRegClient supplies the administrator credentials to Policy Server. Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

If you are installing the Administrative UI as part a new environment, specify the default administrator account (siteminder). If you are installing the UI as an upgrade, specify any administrator account with super user permissions in the policy store.

**Note**: If you are upgrading from r12.0 SP1 and you do not have a super user account in the policy store, create an account using the smreg utility.

## Review the Installation Prerequisites

Consider the following items before you install the Administrative UI:

- Install the Administrative UI using the installation media on the Technical Support site. For a list of installation media names, see the *Policy Server Release Notes*.
- There is one installation zip for the prerequisite installer and one for the Administrative UI installer. Extract all executables to the same location.
- The Administrative UI installation zip contains a layout.properties file. If you move the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.
- Depending on your permissions, run the following command to add executable permissions to the directory that contains the installation media:

  ```
  chmod -R+x directory
  ```

  *directory* specifies the directory that contains the installation media.
- If you execute the Administrative UI installer across different subnets, it can crash. Install the Administrative UI directly on the host system.

# Install the Administrative UI

Install the Administrative UI and prerequisite components to provide a console for management.

These instructions are for GUI and Console mode installations. The steps for the two modes are the same, with the following exceptions for Console Mode:

- Console mode instructions include the command **-i console**.
- Select an option by entering a corresponding number.

- Press ENTER after each step to proceed through the process.
- Type BACK to visit the previous step.

**Follow these steps:**

1. Exit all applications that are running.
2. Open a shell and navigate to the prerequisite installation media.
3. Enter *one* of the following commands:
   **GUI mode:**

   ```
   ./prerequisite_installation_media
   ```

   **Console mode:**

   ```
   ./prerequisite_installation_media -i console
   ```

   *prerequisite_installation_media* specifies the Administrative UI prerequisite installer executable.
   The installer starts.
4. Click Install.
   The required components are installed. The prerequisite installer prompts you to run the Administrative UI installer.
5. Enter *one* of the following command:
   **GUI mode:**

   ```
   ./installation_media
   ```

   **Console mode:**

   ```
   ./installation_media -i console
   ```

   *installation_media* specifies the Administrative UI installer executable.
   The Administrative UI installer starts.
6. Follow the prompts and click Install.
   The Administrative UI is installed.

# Start the Embedded Application Server

Start the application server that is included with the stand-alone installation.

**Follow these steps:**

1. Log in to the Administrative UI host system.
2. Navigate to *install_home*/CA/siteminder/adminui/bin.
3. Execute the following command:

```
run.sh
```

The application server is started.

# Register the Administrative UI

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

**Follow these steps:**

1. Complete *one* of the following steps:

   - (Recommended) Open a web browser and go to the following location to register the Administrative UI over SSL:
     https://*host*:8443/iam/siteminder/adminui

   - Open a browser and go to the following location:
     http://*host*:8080/iam/siteminder/adminui
     *host* specifies the fully qualified Administrative UI host system name.

     If the host system does not have a web browser, you can remotely access the login screen.

   The Administrative UI login screen appears.

2. In the login screen, enter **siteminder** in **User Name**.

3. Enter the siteminder account password in Password.

4. **Note**: If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password**.

5. Enter the fully qualified Policy Server host name in Server.

6. Type the fully qualified Policy Server host name in the Server field. Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.

   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

The Administrative UI opens and is registered with the Policy Server.

Configure additional Policy Server connections for the Administrative UI, or proceed to install an agent.

If you encountered any installation issues, use the following log files to troubleshoot the issues:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  **Default Location**: *administrative_ui_home*\adminui\install_config_info

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  **Default Location**: *administrative_ui_home*\adminui\install_config_info

# (Optional) Change the Connection to the Administrative UI from HTTP to HTTPS

If you used HTTP to register the Administrative UI with Policy Server, you can change the connection type from HTTP to HTTPS. Modify the context.xml file of the embedded JBoss application server to enable secure cookies.

**Follow these steps:**

1. Shut down the application server.

2. Navigate to the following location: user_console.war\WEB-INF

3. Open the **context.xml** file.

4. Add the secure="true" attribute to the <SessionCookie> tag. For example:

   ```
   <SessionCookie secure="true" httpOnly="true" />
   ```

5. Save and close the file.

6. Restart the application server.

## (Optional) Configure an External Administrator Store for UI Administrators

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# Install and Register the Administrative UI on a JBoss Application Server (Windows)

**Contents**



## Prepare for the Administrative UI Installation

Prepare for Administrative UI installation on an existing JBoss/Windows infrastructure by performing the following procedures:

- Verify that the Windows host meets system requirements (see page 404).
- Locate the platform support matrix (see page 404).
- Locate the installation media (see page 387).

- Prepare JBoss for Administrative UI installation (see page ).
- Gather JBoss Information (see page 405).

## Verify Windows System Requirements

Verify that the Windows system that hosts the application server meets the following minimum system requirements. These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**—x86 or x64, 1.2 GHz or better.
- **Memory**—1 GB of system RAM. We recommend 2 GB.
- **Available disk space**—540 MB.
- **Temp directory space**—3 GB.
- **JDK**—A supported JDK is present.
- **Screen resolution**—1024 x 768 or higher resolution with 256 colors or better to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Disable HDScanner on the JBoss Server

JBoss hot-deployable services are services that you can add or removed from the server while it is running. To install the Administrative UI, disable the HDScanner service.

**Follow these steps:**

1. Navigate to *jboss_home*\server*server_profile*\deploy.
   - *jboss_home*
     Specifies the JBoss installation path.
   - *server_profile*
     Specifies the name of the server profile deployed in the application server.
     **Example:** default
2. Remove the following file to disable the service:
   hdscanner-jboss-beans.xml

## Gather JBoss Information

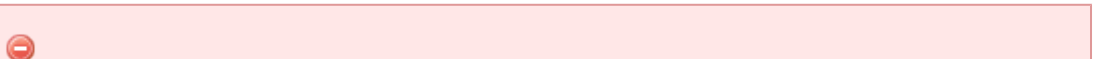
Gather the following information about JBoss before installing and registering the Administrative UI:

- **JBoss installation folder**
  The path to the folder where JBoss is installed.
- **JBoss URL**
  The fully qualified URL of the JBoss host system.
- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.



**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI on an Existing Application Server

Complete the following procedures to install the Administrative UI on your existing Application Server:

1. Review prerequisite information.
2. Install the Administrative UI.
3. Troubleshoot the Administrative UI installation.

For high availability, you can install and configure additional UIs, but this is optional.

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- Run the installer from the Administrative UI host system. Do not run the installer from a mapped network share or UNC path.
- The Administrative installation zip contains a layout.properties file at the same level as the installation media. If you move the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

## Install the Administrative UI on a Windows System

Install the Administrative UI to your existing application server to provide a management console for all tasks that are related to access control, reporting, and policies.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the installation media.
3. Double-click *installation_media*.
   *installation_media* specifies the Administrative UI installation executable.
   The installer starts.
   **Note:** For a list of installation media names, see the Policy Server Release Notes.
4. Based on the information you gathered, enter the required values for the installation.

5. Review the installation settings and click Install.
   The Administrative UI is installed.

> **Note:** You cannot use the Administrative UI to manage your environment until you have registered it with a Policy Server.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the Administrative UI registration window (see page ).
2. Start the application server (see page 409).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window

Reset the Administrative UI registration window if you are installing Administrative UI after 24 hours of performing *one* of the following steps:

- Configured a policy store during the Policy Server installation.

- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

**Follow these steps:**

1. Log in to the Policy Server host system.

2. Run the following command:

   ```
   XPSRegClient siteminder_administrator[:passphrase] -adminui-setup -t timeout -r
   retries -c comment -cp
   -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   ***siteminder_administrator***

   Defines the administrator.

   ***passphrase***

   Defines the password for the administrator account.

   ***-adminui-setup***

   Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   ***-t timeout***

   (Optional) Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.

   Default: 1440 (24 hours)

   Minimum: 1

   Maximum: 1440 (24 hours)

   **-r retries**

   (Optional) Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.

   Default: 1

   Maximum: 5

   **-c comment**

   (Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

   **-cp**

(Optional) Specifies that the registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-l log_path**

(Optional) Specifies where the registration log file must be exported.

Default: siteminder_home\log

***-e error_path***

(Optional) Sends exceptions to the specified path.

Default: stderr

**-vT**

(Optional) Sets the verbosity level to TRACE.

**-vI**

(Optional) Sets the verbosity level to INFO.

**-vW**

(Optional) Sets the verbosity level to WARNING.

**-vE**

(Optional) Sets the verbosity level to ERROR.

**-vF**

(Optional) Sets the verbosity level to FATAL.

XPSRegClient supplies the administrator credentials to Policy Server. Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the JBoss Application Server (Windows)

Start the JBoss Application Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Follow these steps:**

1. On the JBoss host system, navigate to *jboss_home*\bin from a command prompt.

   - *jboss_home*
     Specifies the JBoss installation path.

2. Enter the following command:

   ```
   run.bat
   ```

   The application server is started.

To stop the JBoss server for any reason:

1. From the Administrative UI host system, open the Start Task Engine Command prompt.
2. Enter the following keyboard combination:

   ```
   Ctrl+c
   ```

   The application server stops.

## Register the Administrative UI (Windows)

Register the Administrative UI with a Policy Server so you can begin managing your environment.

**Follow these steps:**

1. Open the Administrative UI from the Administrative UI shortcut.
   The shortcut registers the Administrative UI over SSL. If you do not have access to the shortcut, open a web browser and go to the following location:

   ```
   https://host:port/iam/siteminder/adminui
   ```

   *host* specifies the fully qualified Administrative UI host system name.
   *port* specifies the port on which the application server listens for HTTP requests.

   **Note:** A self–signed certificate that is valid for ten years is created and used for the connection. The certificate is created with an RSA 2048 key strength.

   The Administrative UI login screen displays.

2. Enter **siteminder** in the User Name field.
3. Type the super user account password in the Password field.

**Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.
   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

## Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configure an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store)

# Install the Administrative UI on a JBoss Application Server (UNIX)

**Contents**



## Prepare for the Administrative UI Installation

To prepare for Administrative UI installation on an existing JBoss Application Server on a UNIX system, first perform the following preparatory procedures:

- Verify UNIX system requirements.
- Locate the platform support matrix.
- Locate the installation media.
- Prepare JBoss for Administrative UI installation.
- Gather JBoss Information.

## Verify UNIX System Requirements

Verify that the UNIX system meets the following minimum system requirements. These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**
  - Solaris—UltraSparc, 440 MHz or higher.
  - Red Hat Linux—x86 or x64, 700 MHz or higher.
    The Red Hat 6 operating system relies on entropy for performance. Increase entropy before installing the component. Without sufficient entropy, the installation can take an exceedingly long time to complete. Use the following command to set a symbolic link:

    ```
    mv /dev/random /dev/random.org
    ln -s /dev/urandom /dev/random
    ```

- **Memory**—1 GB of system RAM. We recommend 2 GB.
- **Available disk space**—540 MB.
- **Temp directory space**—3 GB.
- **JDK**—A supported JDK is present.

  **Note:** If your application server runs on a Red Hat Linux operating system, install unlimited cryptography jar files for an IBM JDK when installing the Administrative UI.

- **Screen resolution**—1024 x 768 or higher resolution with a minimum of 256 colors to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Prepare JBoss for Administrative UI Installation

JBoss hot-deployable services are services that you can add or removed from the server while it is running. To install the Administrative UI, disable the HDScanner service.

**Follow these steps:**

1. Navigate to *jboss_home*\server*server_profile*\deploy.
   - *jboss_home*
     Specifies the JBoss installation path.
   - *server_profile*
     Specifies the name of the server profile deployed in the application server.
     **Example:** default
2. Remove the following file to disable the service:
   hdscanner-jboss-beans.xml

## Gather JBoss Information

Gather the following information about JBoss before installing and registering the Administrative UI:

- **JBoss installation folder**
  The path to the folder where JBoss is installed.

- **JBoss URL**
  The fully qualified URL of the JBoss host system.

- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI (UNIX)

Complete the following procedures to install the Administrative UI on your existing application server:

1. Review prerequisite information.
2. Install the Administrative UI on a UNIX System.
3. Troubleshoot the Administrative UI installation.

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- The installation zip contains a layout.properties file at the same level as the installation media. If you moved the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

- Run the following command if you do not have execute permission for the directory that contains the installation media:

  ```
  chmod -R+x directory
  ```

  - **directory**
    Specifies the directory that contains the installation media.

- The user installing the Administrative UI must have read/write permissions for the directory to which the application server is installed.

- If you execute the Administrative UI installer across different subnets, it can crash. Install the Administrative UI directly on the host system.

## Required Linux Libraries

Certain library files are required for components operating on Linux operating environments. Failure to install the correct libraries can cause the following error:

```
java.lang.UnsatisfiedLinkError
```

If you are installing, configuring, or upgrading a Linux version of this component, the following libraries are required on the host system:

- **Red Hat 5.x**
  compat–gcc-34-c++-3.4.6-*patch_version*.I386

- **Red Hat 6.x (32-bit)**
  libstdc++-4.4.6-3.el6.i686.rpm
  To have the appropriate 32-bit C run–time library for your operating environment, install the previous rpm.

- **Red Hat 6.x (64-bit)**
  libXau-1.0.5-1.el6.i686.rpm
  libxcb-1.5-1.el6.i686.rpm
  libstdc++-4.4.6-4.el6.i686.rpm
  compat-db42-4.2.52-15.el6.i686.rpm
  compat-db43-4.3.29-15.el6.i686.rpm
  libX11-1.3-2.el6.i686.rpm
  libXrender-0.9.5-1.el6.i686.rpm
  libexpat.so.1 (provided by expat-2.0.1-11.el6_2.i686.rpm)
  libfreetype.so.6 (provided by freetype-2.3.11-6.el6_2.9.i686.rpm)
  libfontconfig.so.1 (provided by fontconfig-2.8.0-3.el6.i686.rpm)
  libICE-1.0.6-1.el6.i686.rpm
  libuuid-2.17.2-12.7.el6.i686.rpm
  libSM-1.1.0-7.1.el6.i686.rpm
  libXext-1.1-3.el6.i686.rpm
  compat-libstdc++-33-3.2.3-69.el6.i686.rpm
  compat-db-4.6.21-15.el6.i686.rpm
  libXi-1.3-3.el6.i686.rpm

libXtst-1.0.99.2-3.el6.i686.rpm
libXft-2.1.13-4.1.el6.i686.rpm
libXt-1.0.7-1.el6.i686.rpm
libXp-1.0.0-15.1.el6.i686.rpm

## Install the Administrative UI on a UNIX System

Install the Administrative UI to your existing application server. The UI provides a management console for all tasks that are related to access control, reporting, and policy analysis.

These instructions are for GUI and Console mode installations. The steps for the two modes are the same, with the following exceptions for Console Mode:

- Console mode instructions include the command **-i console**.
- Select an option by entering a corresponding number.
- Press ENTER after each step to proceed through the process.
- Type BACK to visit the previous step.

**Follow these steps:**

1. Exit all applications that are running in the foreground.
2. Open a shell and navigate to the installation media.
3. Enter one of the following commands:
   - **GUI mode**
     ./*installation_media*
   - **Console mode**
     ./*installation_media* -i console

   *installation_media*
   Specifies the Administrative UI installation binary.
   The installer starts.
4. Based on the information you gathered, enter the required values.
5. Review the installation settings and click Install (GUI) or press Enter (Console).
   The installation begins.
6. When the installation is complete, click Done (GUI) or press Enter (Console).
7. Reboot the system.
   The Administrative UI is installed.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the Administrative UI registration window (see page ).
2. Start the JBoss Application Server (see page 419).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window (UNIX Systems)

If you completed either of the following actions more than 24 hours before installing the Administrative UI, reset the Administrative UI registration period.

- Configured the policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

Verify that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server.
2. Run the following command:

```
XPSRegClient ps_administrator[:passphrase] -adminui-setup -t timeout -r retries
-c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

*ps_administrator*
Specifies a Policy Server administrator account. If you are installing the Administrative UI as part of a new environment, specify the default administrator account (siteminder). If you are upgrading your environment, specify any administrator account with super user permissions in the policy store.

*passphrase*
Specifies the password for the Policy Server administrator account.
If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm it.

-adminui-setup
Specifies that the Administrative UI is being registered with a Policy Server for the first time.

-t *timeout*
(Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and you create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
**Unit of measure:** minutes
**Default:** 1440 (24 hours)
**Minimum:** 1
**Maximum:** 1440 (24 hours)

-r *retries*
(Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
**Default:** 1
**Maximum:** 5

-c *comment*
(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-cp
(Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-l *log path*
(Optional) Specifies where the registration log file must be exported.
**Default:***siteminder_home/*log
*siteminder_home* specifies *t*he Policy Server installation path.

-e *error path*
(Optional) Sends exceptions to the specified path.
**Default:** stderr

-vT
(Optional) Sets the verbosity level to TRACE.

-vI
(Optional) Sets the verbosity level to INFO.

-vW
(Optional) Sets the verbosity level to WARNING.

-vE
(Optional) Sets the verbosity level to ERROR.

-vF
(Optional) Sets the verbosity level to FATAL.Press Enter.

3. XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the JBoss Application Server (UNIX)

Start the JBoss Application Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Follow these steps:**

1. On the JBoss host system, navigate to *jboss_home/*bin from a command prompt.

   - **jboss_home**
     Specifies the JBoss installation path.

2. Type the following command and press Enter:

   ```
   run.sh
   ```

   The application server is started.

To stop the JBoss server for any reason:

1. From a command window, navigate to *jboss_home*/bin.
   *jboss_home*
   Specifies the JBoss installation path.

2. Enter the following command:

   ```
   ./shutdown.sh
   ```

   The application server stops.

## Register the Administrative UI (UNIX)

Register the Administrative UI with a Policy Server so you can use it for managing your environment.

**Follow these steps:**

1. Open a web browser and go to the following location:

   `host:port/iam/siteminder/adminui`

   If the host system does not have a web browser, you can remotely access the login screen.

   - *host*
     Specifies the fully qualified Administrative UI host system name.

   - *port*
     Specifies the port on which the application server listens for HTTP requests.

   The Administrative UI login screen appears.

2. Enter **siteminder** in the User Name field.

3. Type the super user account password in the Password field.

   **Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.

   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

# Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# Install and Register the Administrative UI on a WebLogic Server (Windows)

**Contents**



# Prepare for the Administrative UI Installation

To prepare for Administrative UI installation on an existing WebLogic infrastructure on a Windows system, first perform the following preparatory procedures:

- Verify that the Windows host meets system requirements (see page 404).
- Locate the platform support matrix (see page 404).
- Locate the installation media (see page 387).
- Prepare WebLogic Server for Administrative UI installation (see page ).
- Gather WebLogic Information (see page 423).

## Verify Windows System Requirements

Verify that the Windows system that hosts the application server meets the following minimum system requirements. These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**—x86 or x64, 1.2 GHz or better.

- **Memory**—1 GB of system RAM. We recommend 2 GB.
- **Available disk space**—540 MB.
- **Temp directory space**—3 GB.
- **JDK**—A supported JDK is present.
- **Screen resolution**—1024 x 768 or higher resolution with 256 colors or better to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Prepare WebLogic Server for Administrative UI Installation

Prepare the WebLogic Server for Administrative UI installation.

**Follow these steps:**

1. Create a WebLogic domain using the Configuration Wizard that is part of the WebLogic installation and do the following steps:

    a. Note the name of the domain, which is required when installing the Administrative UI.

    b. Select the Basic WebLogic Server Domain template.

    c. Verify that the JAVA_HOME variable is set to the path for the required Java environment in the setDomainEnv.cmd/ .sh file. This file is located in *web_logic_home* \user_projects\domains*weblogic_domain*\bin.

        - **web_logic_home**
          Specifies the WebLogic server installation path.

        - **weblogic_domain**
          Specifies the name of the WebLogic domain you created.

2. Confirm that the WebLogic server is running and that you can access the WebLogic console at http://*server.domain.port*/console.
   **Example:** http://myserver.example.com:7001/console

3. In the WebLogic console, under Domain Configurations, select the domains link and verify that the WebLogic domain you created appears in the list of existing domains.

4. Shut down the application server in preparation for the Administrative UI installation.

## Gather WebLogic Information

Gather the following information before installing and registering the Administrative UI:

- **WebLogic binary folder**
  The path to the WebLogic installation directory.

- **WebLogic domain folder**
  The path to the WebLogic domain you created for the Administrative UI.

- **WebLogic server name**
  The name of the WebLogic server on which the WebLogic domain is configured.

- **Application server URL and port**
  The fully qualified URL of the WebLogic host system.

- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI on the Application Server

Complete the following procedures to install the Administrative UI on your existing Application Server:

1. Review prerequisite information.
2. Install the Administrative UI.
3. Troubleshoot the Administrative UI installation.

For high availability, you can install and configure additional UIs, but this is optional.

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- Run the installer from the Administrative UI host system. Do not run the installer from a mapped network share or UNC path.
- The Administrative installation zip contains a layout.properties file at the same level as the installation media. If you move the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

## Install the Administrative UI on a Windows System

Install the Administrative UI to your existing application server to provide a management console for all tasks that are related to access control, reporting, and policies.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the installation media.

3. Double-click *installation_media*.
   *installation_media* specifies the Administrative UI installation executable.
   The installer starts.
   **Note:** For a list of installation media names, see the Policy Server Release Notes.

4. Based on the information you gathered, enter the required values for the installation.

5. Review the installation settings and click Install.
   The Administrative UI is installed.

> **Note:** You cannot use the Administrative UI to manage your environment until you have registered it with a Policy Server.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the registration window (see page ).
2. Start the application server (see page 427).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window

Reset the Administrative UI registration window if you are installing Administrative UI after 24 hours of performing *one* of the following steps:

- Configured a policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder_administrator[:passphrase] -adminui-setup -t timeout -r
   retries -c comment -cp
   -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   ***siteminder_administrator***

   Defines the administrator.

   ***passphrase***

   Defines the password for the administrator account.

   ***-adminui-setup***

   Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   ***-t timeout***

   (Optional) Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.

   Default: 1440 (24 hours)

   Minimum: 1

   Maximum: 1440 (24 hours)

   **-r retries**

   (Optional) Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.

   Default: 1

   Maximum: 5

   **-c comment**

(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-cp**

(Optional) Specifies that the registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-l log_path**

(Optional) Specifies where the registration log file must be exported.

Default: siteminder_home\log

***-e error_path***

(Optional) Sends exceptions to the specified path.

Default: stderr

**-vT**

(Optional) Sets the verbosity level to TRACE.

**-vI**

(Optional) Sets the verbosity level to INFO.

**-vW**

(Optional) Sets the verbosity level to WARNING.

**-vE**

(Optional) Sets the verbosity level to ERROR.

**-vF**

(Optional) Sets the verbosity level to FATAL.

XPSRegClient supplies the administrator credentials to Policy Server. Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the WebLogic Application Server (Windows)

Start the WebLogic Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Follow these steps:**

1. On the WebLogic host system, navigate to *domains*\bin from a command prompt.
   - **domains**
     Specifies the path of the WebLogic domain you created for the Administrative UI.
     **Example:** C:\bea\user_projects\domains\mydomain
2. Type the following command and press Enter:

   ```
   ./startWebLogic.cmd
   ```

   The application server is started.

To stop the WebLogic server for any reason:

1. From the Administrative UI host system, open the Start Task Engine Command prompt.
2. Enter the following keyboard combination:

   ```
   Ctrl+c
   ```

   The application server stops.

## Register the Administrative UI (Windows)

Register the Administrative UI with a Policy Server so you can begin managing your environment.

**Follow these steps:**

1. Open the Administrative UI from the Administrative UI shortcut.
   The shortcut registers the Administrative UI over SSL. If you do not have access to the shortcut, open a web browser and go to the following location:

   ```
   https://host:port/iam/siteminder/adminui
   ```

   *host* specifies the fully qualified Administrative UI host system name.
   *port* specifies the port on which the application server listens for HTTP requests.

   **Note:** A self–signed certificate that is valid for ten years is created and used for the connection. The certificate is created with an RSA 2048 key strength.

The Administrative UI login screen displays.

2. Enter **siteminder** in the User Name field.

3. Type the super user account password in the Password field.

   **Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.
   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

# Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store)

# Install and Register the Administrative UI on a WebLogic Server (UNIX)

**Contents**

# Prepare for the Administrative UI Installation

To prepare for Administrative UI installation on an existing WebLogic infrastructure on a UNIX system, first perform the following preparatory procedures:

- Verify UNIX system requirements (see page 412).
- Locate the platform support matrix (see page 404).
- Locate the installation media (see page 387).
- Prepare WebLogic Server for Administrative UI installation (see page ).
- Gather WebLogic Information (see page 423).

## Verify UNIX System Requirements

Verify that the UNIX system meets the following minimum system requirements. These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**
  - Solaris—UltraSparc, 440 MHz or higher.
  - Red Hat Linux—x86 or x64, 700 MHz or higher.
    The Red Hat 6 operating system relies on entropy for performance. Increase entropy before installing the component. Without sufficient entropy, the installation can take an exceedingly long time to complete. Use the following command to set a symbolic link:

    ```
    mv /dev/random /dev/random.org
    ln -s /dev/urandom /dev/random
    ```

- **Memory**—1 GB of system RAM. We recommend 2 GB.
- **Available disk space**—540 MB.
- **Temp directory space**—3 GB.

- **JDK**—A supported JDK is present.

  **Note:** If your application server runs on a Red Hat Linux operating system, install unlimited cryptography jar files for an IBM JDK when installing the Administrative UI.

- **Screen resolution**—1024 x 768 or higher resolution with a minimum of 256 colors to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Prepare WebLogic Server for Administrative UI Installation

Prepare WebLogic Server for Administrative UI installation.

**Follow these steps:**

1. Create a WebLogic domain using the Configuration wizard that is part of the WebLogic installation and do the following steps:

a. Note the name of the domain, which is required when installing the Administrative UI.

b. Select the Basic WebLogic Server Domain template.

c. Verify that the JAVA_HOME variable is set to the path for the required Java environment in the setDomainEnv.cmd/ .sh file. This file is located in *web_logic_home/* user_projects/domains/*weblogic_domain*\bin.

- **web_logic_home**
  Specifies the WebLogic server installation path.

- **weblogic_domain**
  Specifies the name of the WebLogic domain you created.

2. Confirm that the WebLogic server is running and that you can access the WebLogic console at http://*server.domain.port*/console.
   **Example:** http://myserver.example.com:7001/console

3. In the WebLogic console, under Domain Configurations, select the domains link and verify that the WebLogic domain you created appears in the list of existing domains.

4. Shut down the application server in preparation for the Administrative UI installation.

## Gather WebLogic Information

Gather the following information before installing and registering the Administrative UI:

- **WebLogic binary folder**
  The path to the WebLogic installation directory.

- **WebLogic domain folder**
  The path to the WebLogic domain you created for the Administrative UI.

- **WebLogic server name**
  The name of the WebLogic server on which the WebLogic domain is configured.

- **Application server URL and port**
  The fully qualified URL of the WebLogic host system.

- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the

XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI (UNIX)

Complete the following procedures to install the Administrative UI on your existing application server:

1. Review prerequisite information.
2. Install the Administrative UI on a UNIX System.
3. Troubleshoot the Administrative UI installation.

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- The installation zip contains a layout.properties file at the same level as the installation media. If you moved the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.
- Run the following command if you do not have execute permission for the directory that contains the installation media:

  ```
  chmod -R+x directory
  ```

  - **directory**
    Specifies the directory that contains the installation media.
- The user installing the Administrative UI must have read/write permissions for the directory to which the application server is installed.
- If you execute the Administrative UI installer across different subnets, it can crash. Install the Administrative UI directly on the host system.

## Required Linux Libraries

Certain library files are required for components operating on Linux operating environments. Failure to install the correct libraries can cause the following error:

```
java.lang.UnsatisfiedLinkError
```

If you are installing, configuring, or upgrading a Linux version of this component, the following libraries are required on the host system:

- **Red Hat 5.x**
  compat–gcc-34-c++-3.4.6-*patch_version*.I386

- **Red Hat 6.x (32-bit)**
  libstdc++-4.4.6-3.el6.i686.rpm
  To have the appropriate 32-bit C run–time library for your operating environment, install the previous rpm.

- **Red Hat 6.x (64-bit)**
  libXau-1.0.5-1.el6.i686.rpm
  libxcb-1.5-1.el6.i686.rpm
  libstdc++-4.4.6-4.el6.i686.rpm
  compat-db42-4.2.52-15.el6.i686.rpm
  compat-db43-4.3.29-15.el6.i686.rpm
  libX11-1.3-2.el6.i686.rpm
  libXrender-0.9.5-1.el6.i686.rpm
  libexpat.so.1 (provided by expat-2.0.1-11.el6_2.i686.rpm)
  libfreetype.so.6 (provided by freetype-2.3.11-6.el6_2.9.i686.rpm)
  libfontconfig.so.1 (provided by fontconfig-2.8.0-3.el6.i686.rpm)
  libICE-1.0.6-1.el6.i686.rpm
  libuuid-2.17.2-12.7.el6.i686.rpm
  libSM-1.1.0-7.1.el6.i686.rpm
  libXext-1.1-3.el6.i686.rpm
  compat-libstdc++-33-3.2.3-69.el6.i686.rpm
  compat-db-4.6.21-15.el6.i686.rpm
  libXi-1.3-3.el6.i686.rpm
  libXtst-1.0.99.2-3.el6.i686.rpm
  libXft-2.1.13-4.1.el6.i686.rpm
  libXt-1.0.7-1.el6.i686.rpm
  libXp-1.0.0-15.1.el6.i686.rpm

## Install the Administrative UI on a UNIX System

Install the Administrative UI to your existing application server. The UI provides a management console for all tasks that are related to access control, reporting, and policy analysis.

These instructions are for GUI and Console mode installations. The steps for the two modes are the same, with the following exceptions for Console Mode:

- Console mode instructions include the command **-i console**.

- Select an option by entering a corresponding number.

- Press ENTER after each step to proceed through the process.

- Type BACK to visit the previous step.

**Follow these steps:**

1. Exit all applications that are running in the foreground.

2. Open a shell and navigate to the installation media.

3. Enter one of the following commands:

   - **GUI mode**
     *./installation_media*

   - **Console mode**
     *./installation_media* -i console

   *installation_media*
   Specifies the Administrative UI installation binary.
   The installer starts.

4. Based on the information you gathered, enter the required values.

5. Review the installation settings and click Install (GUI) or press Enter (Console).
   The installation begins.

6. When the installation is complete, click Done (GUI) or press Enter (Console).

7. Reboot the system.
   The Administrative UI is installed.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the Administrative UI registration window (see page ).
2. Start the WebLogic Server (see page 437).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window (UNIX Systems)

If you completed either of the following actions more than 24 hours before installing the Administrative UI, reset the Administrative UI registration period.

- Configured the policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

Verify that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server.
2. Run the following command:

   ```
   XPSRegClient ps_administrator[:passphrase] -adminui-setup -t timeout -r retries
   -c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   *ps_administrator*
   Specifies a Policy Server administrator account. If you are installing the Administrative UI as part of a new environment, specify the default administrator account (siteminder). If you are upgrading your environment, specify any administrator account with super user permissions in the policy store.

   *passphrase*
   Specifies the password for the Policy Server administrator account.
   If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm it.

   -adminui-setup
   Specifies that the Administrative UI is being registered with a Policy Server for the first time.

   -t *timeout*
   (Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and you create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.

**Unit of measure:** minutes
**Default:** 1440 (24 hours)
**Minimum:** 1
**Maximum:** 1440 (24 hours)

-r *retries*
(Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
**Default:** 1
**Maximum:** 5

-c *comment*
(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-cp
(Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-l *log path*
(Optional) Specifies where the registration log file must be exported.
**Default:***siteminder_home/*log
*siteminder_home* specifies *t*he Policy Server installation path.

-e *error path*
(Optional) Sends exceptions to the specified path.
**Default:** stderr

-vT
(Optional) Sets the verbosity level to TRACE.

-vI
(Optional) Sets the verbosity level to INFO.

-vW
(Optional) Sets the verbosity level to WARNING.

-vE
(Optional) Sets the verbosity level to ERROR.

-vF
(Optional) Sets the verbosity level to FATAL.Press Enter.

3. XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the WebLogic Server (UNIX)

Start the WebLogic Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Follow these steps:**

1. On the WebLogic host system, navigate to *domains/*bin from a command prompt.
   - **domains**
     Specifies the path of the WebLogic domain you created for the Administrative UI.
2. Type the following command and press Enter:

   ```
   ./startWebLogic.sh
   ```

   The application server is started.

To stop the WebLogic server for any reason:

1. From a command window, navigate to *domains*/bin.
   *domains*
   Specifies the path of the WebLogic domain you created for the Administrative UI.
2. Enter the following command:

   ```
   ./stopWebLogic.sh
   ```

   The application server stops.

## Register the Administrative UI (UNIX)

Register the Administrative UI with a Policy Server so you can use it for managing your environment.

**Follow these steps:**

1. Open a web browser and go to the following location:

   *host:port*/iam/siteminder/adminui

   If the host system does not have a web browser, you can remotely access the login screen.
   - *host*
     Specifies the fully qualified Administrative UI host system name.
   - *port*
     Specifies the port on which the application server listens for HTTP requests.

   The Administrative UI login screen appears.
2. Enter **siteminder** in the User Name field.
3. Type the super user account password in the Password field.

> **Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.
   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

# Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# Install and Register an Administrative UI on a WebSphere Application Server (Windows)

**Contents**

# Prepare for the Administrative UI Installation

To prepare for Administrative UI installation on an existing WebSphere Application Server on a Windows system, first perform the following preparatory procedures:

- Verify that the Windows system requirements (see page 440).
- Locate the platform support matrix (see page 404).
- Locate the installation media (see page 387).
- Prepare WebSphere Application Server for Administrative UI installation (see page ).
- Gather WebSphere information (see page 441).

## Verify Windows System Requirements

Verify that the Windows system meets the following minimum system requirements These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**—x86 or x64, 1.2 GHz or higher.
- **Memory**—2 GB of system RAM. We recommend 2 GB.
- **Available disk space**—2 GB.
- **Temp directory space**—3 GB.
- **JDK**—A supported JDK is present.
- **Screen resolution**—1024 x 768 or higher resolution with a minimum of 256 colors to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.

2. In the top field, start typing the product name you are interested in then select the appropriate entry.

3. Mouse over the Knowledge Center option and click **Product Pages**.

4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.

5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.

6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Prepare WebSphere for Administrative UI Installation

Prepare the WebSphere Application Server for Administrative UI installation.

**Follow these steps:**

1. Verify that the Server and Client option was selected during WebSphere installation.

2. Verify that WebSphere is working by doing the following steps:

    a. Enter http://<*fqdn:port*>/snoop to verify that WebSphere is installed correctly.
    **Example:** http:MyServer.example.com:9080/snoop.
    If WebSphere is installed correctly, Snoop Servlet—Request Client Information page is displayed in the browser.

    b. Enter http://<*fqdn*>/snoop to verify that the WebSphere application server plug-in is installed correctly.
    **Example:** http://MyServer.example.com/snoop
    If WebSphere is installed correctly, the same Snoop Servlet—Request Client Information page is displayed in the browser.

3. Disable the Administrative Security option.

4. Shut down the application server to prepare for the Administrative UI installation.

## Gather WebSphere Information

Gather the following information about WebSphere before installing and registering the Administrative UI:

- **WebSphere installation folder**
  The full path to the folder in which WebSphere is installed.
- **WebSphere URL**
  The fully qualified URL of the WebSphere host system.
- **Server name**
  The name of the application server.
- **Profile name**
  The name of the profile being used for the Administrative UI.
- **Cell name**
  The name of the cell where the server is located.
- **Node name**
  The name of the node where the server is located.
- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI

Complete the following procedures to install the Administrative UI to your existing WebSphere Application Server:

1. Review prerequisite information (see page 406).

2. Install the Administrative UI.

3. Troubleshoot the Administrative UI installation (see page ).

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- Run the installer from the Administrative UI host system. Do not run the installer from a mapped network share or UNC path.

- The Administrative installation zip contains a layout.properties file at the same level as the installation media. If you move the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

## Install the Administrative UI on a Windows System

Install the Administrative UI to your existing application server to provide a management console for all tasks that are related to access control, reporting, and policies.

**Follow these steps:**

1. Exit all applications that are running.

2. Navigate to the installation media.

3. Double-click *installation_media*.
   *installation_media* specifies the Administrative UI installation executable.
   The installer starts.
   **Note:** For a list of installation media names, see the Policy Server Release Notes.

4. Based on the information you gathered, enter the required values for the installation.

5. Review the installation settings and click Install.
   The Administrative UI is installed.

**Note:** You cannot use the Administrative UI to manage your environment until you have registered it with a Policy Server.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

**Location:** *administrative_ui_home*\adminui\install_config_info
*administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the Administrative UI registration window (see page ).
2. Start the application server (see page 446).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window

Reset the Administrative UI registration window if you are installing Administrative UI after 24 hours of performing *one* of the following steps:

- Configured a policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

**Follow these steps:**

1. Log in to the Policy Server host system.
2. Run the following command:

   ```
   XPSRegClient siteminder_administrator[:passphrase] -adminui-setup -t timeout -r
   retries -c comment -cp
   -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   ***siteminder_administrator***

   Defines the administrator.

   ***passphrase***

Defines the password for the administrator account.

***-adminui-setup***

Specifies that the Administrative UI is being registered with a Policy Server for the first time.

***-t timeout***

(Optional) Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.

Default: 1440 (24 hours)

Minimum: 1

Maximum: 1440 (24 hours)

**-r retries**

(Optional) Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.

Default: 1

Maximum: 5

**-c comment**

(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-cp**

(Optional) Specifies that the registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

**-l log_path**

(Optional) Specifies where the registration log file must be exported.

Default: siteminder_home\log

***-e error_path***

(Optional) Sends exceptions to the specified path.

Default: stderr

**-vT**

(Optional) Sets the verbosity level to TRACE.

**-vI**

(Optional) Sets the verbosity level to INFO.

**-vW**

(Optional) Sets the verbosity level to WARNING.

**-vE**

(Optional) Sets the verbosity level to ERROR.

**-vF**

(Optional) Sets the verbosity level to FATAL.

XPSRegClient supplies the administrator credentials to Policy Server. Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the WebSphere Application Server (Windows)

Start the WebSphere Application Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Follow these steps:**

1. On the WebSphere host system, navigate to *profile*\bin from a command prompt.
   *profile*
   Specifies the path of the WebSphere profile name you created for the Administrative UI.
   Example:
   C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSvr01\bin

2. Type the following command and press Enter:

   ```
   startServer.batidentifier
   ```

*identifier*
Specifies the identifier for the WebSphere installation.
Example:

```
startServer.bat Server1
```

The application server is started.

To stop the WebSphere server for any reason:

1. From the Administrative UI host system, open a command prompt.

2. Navigate to *profile*\bin.
   *profile*
   Specifies the path of the WebSphere profile name you created for the Administrative UI.
   Example:
   C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSvr01\bin

3. Enter the following command:

   ```
   stopServer.bat identifier
   ```

   *identifier*
   Specifies the identifier for the WebSphere installation.

   **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

   Example:

   ```
   stopServer.bat Server1
   ```

   The application server stops.

## Register the Administrative UI (Windows)

Register the Administrative UI with a Policy Server so you can begin managing your environment.

**Follow these steps:**

1. Open the Administrative UI from the Administrative UI shortcut.
   The shortcut registers the Administrative UI over SSL. If you do not have access to the shortcut, open a web browser and go to the following location:

   ```
   https://host:port/iam/siteminder/adminui
   ```

   *host* specifies the fully qualified Administrative UI host system name.
   *port* specifies the port on which the application server listens for HTTP requests.

> **Note:** A self–signed certificate that is valid for ten years is created and used for the connection. The certificate is created with an RSA 2048 key strength.

The Administrative UI login screen displays.

2. Enter **siteminder** in the User Name field.
3. Type the super user account password in the Password field.

> **Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:
   - You can enter a valid IPv4 address or IPv6 address.
   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

# Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configuring an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# Install and Register the Administrative UI on a WebSphere Application Server (UNIX)

**Contents**

# Prepare for Installation

To prepare for Administrative UI installation on an existing WebSphere infrastructure on a UNIX system, first perform the following preparatory procedures:

- Verify UNIX system requirements (see page 449).
- Locate the platform support matrix (see page 404).
- Locate the installation media (see page 387).
- Prepare WebSphere for Administrative UI installation (see page ).
- Gather WebSphere Information (see page 441).

## Verify UNIX System Requirements

Verify that the UNIX system meets the following minimum system requirements. These recommendations accommodate only the UI. Size your hardware appropriately for all services running on the same system.

- **CPU**
  **Solaris:** UltraSparc, 440 MHz or higher.
  **Red Hat Linux:** x86 or x64, 700 MHz or higher.
  The Red Hat 6 operating system relies on entropy for performance. Increase entropy before installing the component. Without sufficient entropy, the installation can take an exceedingly long time to complete. We Use the following command to set a symbolic link:

  ```
  mv /dev/random /dev/random.org
  ln -s /dev/urandom /dev/random
  ```

- **Memory**—2 GB of system RAM.

- **Available disk space**—2 GB.
- **Temp directory space**—3 GB.
- **JDK**—The required JDK version is installed on the system to which you are installing the Administrative UI.

  **Note:** If your application server runs on a Red Hat Linux operating system, install unlimited cryptography JAR files for an IBM JDK when installing the Administrative UI.

- **Screen resolution**—1024 x 768 or higher resolution with a minimum of 256 colors to view the Administrative UI properly.

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Prepare WebSphere for Administrative UI Installation

Prepare WebSphere Application Server for Administrative UI installation.

**Follow these steps:**

1. Verify that the Server and Client option was selected during WebSphere installation.
2. (Solaris) Verify that the following groups and users are configured for the Embedded Messaging service:
   - Groups: mqm, mqbrkrs
   - Users: mqm, root

   **Note:** For more information, see the IBM documentation.
3. Verify that WebSphere is working by doing the following steps:
   a. Enter http://<*fqdn:port*>/snoop to verify that WebSphere is installed correctly.
   **Example:** http:MyServer.example.com:9080/snoop.
   If WebSphere is installed correctly, Snoop Servlet—Request Client Information page is displayed in the browser.

   b. Enter http://<*fqdn*>/snoop to verify that the WebSphere application server plug-in is installed correctly.
   **Example:** http://MyServer.example.com/snoop
   If WebSphere is installed correctly, the same Snoop Servlet—Request Client Information page is displayed in the browser.
4. Disable the Administrative Security option.
5. Shut down the application server to prepare for the Administrative UI installation.

## Gather WebSphere Information

Gather the following information about WebSphere before installing and registering the Administrative UI:

- **WebSphere installation folder**
  The full path to the folder in which WebSphere is installed.
- **WebSphere URL**
  The fully qualified URL of the WebSphere host system.
- **Server name**
  The name of the application server.
- **Profile name**
  The name of the profile being used for the Administrative UI.
- **Cell name**
  The name of the cell where the server is located.

- **Node name**
  The name of the node where the server is located.

- **JDK**
  The installation location of the required JDK.

## Trusted Relationship between the UI and the Policy Server

The first time you log in to the Administrative UI with the default super user account (siteminder) and password, the UI is registered with the Policy Server. This registration process establishes a trusted relationship between the Administrative UI and a Policy Server. This relationship is required to manage your environment.

The super user account credentials are stored in the policy store. If you configured one of the default policy stores during the Policy Server installation, the installer submits these credentials automatically. If you configure the policy store independent of the Policy Server installation, use the XPSRegClient utility to submit the credentials to the Policy Server (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247). The Policy Server uses these credentials to verify that the registration request from the UI is valid and that the trust relationship can be created.

**Important!** A 24-hour limit exists between the time the super user account credentials are submitted to the policy store and when the administrator logs in to the Administrative UI. If the credentials were set more than 24 hours before the initial log in to the Administrative UI, reset the credentials using the XPSRegClient utility (https://docops.ca.com/display/sm1252sp1/_Reset+the+Administrative+UI+Registration+Window_2018247).

# Install the Administrative UI (UNIX)

Complete the following procedures to install the Administrative UI on your existing application server:

1. Review prerequisite information.
2. Install the Administrative UI on a UNIX System.
3. Troubleshoot the Administrative UI installation.

## Review Prerequisite Information

Consider the following items before you install the Administrative UI:

- The installation zip contains a layout.properties file at the same level as the installation media. If you moved the installation media after extracting the installation zip, move the properties file to the same location or the installation fails.

- Run the following command if you do not have execute permission for the directory that contains the installation media:

  ```
  chmod -R+x directory
  ```

  - **directory**
    Specifies the directory that contains the installation media.

- The user installing the Administrative UI must have read/write permissions for the directory to which the application server is installed.

- If you execute the Administrative UI installer across different subnets, it can crash. Install the Administrative UI directly on the host system.

## Required Linux Libraries

Certain library files are required for components operating on Linux operating environments. Failure to install the correct libraries can cause the following error:

```
java.lang.UnsatisfiedLinkError
```

If you are installing, configuring, or upgrading a Linux version of this component, the following libraries are required on the host system:

- **Red Hat 5.x**
  compat–gcc-34-c++-3.4.6-*patch_version*.I386

- **Red Hat 6.x (32-bit)**
  libstdc++-4.4.6-3.el6.i686.rpm
  To have the appropriate 32-bit C run–time library for your operating environment, install the previous rpm.

- **Red Hat 6.x (64-bit)**
  libXau-1.0.5-1.el6.i686.rpm
  libxcb-1.5-1.el6.i686.rpm
  libstdc++-4.4.6-4.el6.i686.rpm
  compat-db42-4.2.52-15.el6.i686.rpm
  compat-db43-4.3.29-15.el6.i686.rpm
  libX11-1.3-2.el6.i686.rpm
  libXrender-0.9.5-1.el6.i686.rpm
  libexpat.so.1 (provided by expat-2.0.1-11.el6_2.i686.rpm)
  libfreetype.so.6 (provided by freetype-2.3.11-6.el6_2.9.i686.rpm)
  libfontconfig.so.1 (provided by fontconfig-2.8.0-3.el6.i686.rpm)
  libICE-1.0.6-1.el6.i686.rpm
  libuuid-2.17.2-12.7.el6.i686.rpm
  libSM-1.1.0-7.1.el6.i686.rpm
  libXext-1.1-3.el6.i686.rpm
  compat-libstdc++-33-3.2.3-69.el6.i686.rpm
  compat-db-4.6.21-15.el6.i686.rpm
  libXi-1.3-3.el6.i686.rpm

libXtst-1.0.99.2-3.el6.i686.rpm
libXft-2.1.13-4.1.el6.i686.rpm
libXt-1.0.7-1.el6.i686.rpm
libXp-1.0.0-15.1.el6.i686.rpm

## Install the Administrative UI on a UNIX System

Install the Administrative UI to your existing application server. The UI provides a management console for all tasks that are related to access control, reporting, and policy analysis.

These instructions are for GUI and Console mode installations. The steps for the two modes are the same, with the following exceptions for Console Mode:

- Console mode instructions include the command **-i console**.
- Select an option by entering a corresponding number.
- Press ENTER after each step to proceed through the process.
- Type BACK to visit the previous step.

**Follow these steps:**

1. Exit all applications that are running in the foreground.
2. Open a shell and navigate to the installation media.
3. Enter one of the following commands:
   - **GUI mode**
     ./*installation_media*
   - **Console mode**
     ./*installation_media* -i console

   *installation_media*
   Specifies the Administrative UI installation binary.
   The installer starts.
4. Based on the information you gathered, enter the required values.
5. Review the installation settings and click Install (GUI) or press Enter (Console).
   The installation begins.
6. When the installation is complete, click Done (GUI) or press Enter (Console).
7. Reboot the system.
   The Administrative UI is installed.

## Troubleshoot the Administrative UI Installation

Use the following files to troubleshoot the Administrative UI installation:

- Administrative_UI_Prerequisite_Installer_InstallLog.log
  If you used the stand–alone installation option, this log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the prerequisite installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

- CA_ SiteMinder_Administrative_UI_InstallLog.log
  This log lists the number of successes, warnings, non–fatal errors, and errors that occurred during the Administrative UI installation. Individual installation actions are listed with the respective status.

  **Location:** *administrative_ui_home*\adminui\install_config_info
  *administrative_ui_home* specifies the Administrative UI installation path. For example, C:\CA\siteminder or /opt/CA/siteminder

# Register the Administrative UI (UNIX)

Register the Administrative UI before you use it to manage your environment. Registering the Administrative UI creates a trusted connection between the Administrative UI and a Policy Server.

**Follow these steps:**

1. Reset the Administrative UI registration window (see page ).
2. Start the WebSphere Application Server (see page 457).
3. Register the Administrative UI (see page ).

## Reset the Administrative UI Registration Window (UNIX Systems)

If you completed either of the following actions more than 24 hours before installing the Administrative UI, reset the Administrative UI registration period.

- Configured the policy store during the Policy Server installation.
- Used the XPSRegClient utility to submit the super user credentials to the Policy Server.

Verify that the CA Single Sign-On environment variables are set before you use XPSRegClient. If the environment variables are not set, set them manually.

**Follow these steps:**

1. Log in to the Policy Server.
2. Run the following command:

```
XPSRegClient ps_administrator[:passphrase] -adminui-setup -t timeout -r retries
-c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
```

*ps_administrator*
Specifies a Policy Server administrator account. If you are installing the Administrative UI as part of a new environment, specify the default administrator account (siteminder). If you are upgrading your environment, specify any administrator account with super user permissions in the policy store.

*passphrase*
Specifies the password for the Policy Server administrator account.
If you do not specify the passphrase, XPSRegClient prompts you to enter and confirm it.

-adminui-setup
Specifies that the Administrative UI is being registered with a Policy Server for the first time.

-t *timeout*
(Optional) Specifies the allotted time from when you to install the Administrative UI to the time you log in and you create a trusted relationship with a Policy Server. The Policy Server denies the registration request when the timeout value is exceeded.
**Unit of measure:** minutes
**Default:** 1440 (24 hours)
**Minimum:** 1
**Maximum:** 1440 (24 hours)

-r *retries*
(Optional) Specifies how many failed attempts are allowed when you are registering the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
**Default:** 1
**Maximum:** 5

-c *comment*
(Optional) Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-cp
(Optional) Specifies that registration log file can contain multiple lines of comments. The utility prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

-l *log path*
(Optional) Specifies where the registration log file must be exported.
**Default:***siteminder_home/*log
*siteminder_home* specifies *t*he Policy Server installation path.

-e *error path*
(Optional) Sends exceptions to the specified path.
**Default:** stderr

-vT
(Optional) Sets the verbosity level to TRACE.

-vI
(Optional) Sets the verbosity level to INFO.

-vW
(Optional) Sets the verbosity level to WARNING.

-vE
(Optional) Sets the verbosity level to ERROR.

-vF
(Optional) Sets the verbosity level to FATAL.Press Enter.

3. XPSRegClient supplies the Policy Server with the administrator credentials. The Policy Server uses these credentials to verify the registration request when you log in to the Administrative UI for the first time.

## Start the WebSphere Application Server (UNIX)

Start the WebSphere Application Server to start the Administrative UI web application.

**Note:** Starting the application server allows administrators to *access* the Administrative UI; it does not open the Administrative UI directly.

**Follow these steps:**

1. On the WebSphere host system, navigate to *profile/b*in from a command prompt.

   - **profile**
     Specifies the path of the WebSphere profile name you created for the Administrative UI.

2. Type the following command and press Enter:

   ```
   startServer.shidentifier
   ```

   - **identifier**
     Specifies the identifier for the WebSphere installation.
     **Example:** startServer.sh Server1

   The application server is started.

To stop the WebSphere server for any reason:

1. From a command prompt, navigate to *profile*\bin.
   *profile*
   Specifies the path of the WebSphere profile name you created for the Administrative UI
   **Example:**
   C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSvr01\bin

2. Enter the following command:

   ```
   stopServer.shidentifier
   ```

*identifier*
Specifies the identifier for the WebSphere installation.
**Example:** stopServer.sh Server1
The application server is stopped.

## Register the Administrative UI (UNIX)

Register the Administrative UI with a Policy Server so you can use it for managing your environment.

**Follow these steps:**

1. Open a web browser and go to the following location:

   `host:port/iam/siteminder/adminui`

   If the host system does not have a web browser, you can remotely access the login screen.

   - *host*
     Specifies the fully qualified Administrative UI host system name.

   - *port*
     Specifies the port on which the application server listens for HTTP requests.

   The Administrative UI login screen appears.

2. Enter **siteminder** in the User Name field.

3. Type the super user account password in the Password field.

   **Note:** If your super user account password contains dollar-sign ($) characters, replace each instance of the dollar-sign character with $DOLLAR$. For example, if the super user account password is $password, enter **$DOLLAR$password** in the Password field.

4. Type the fully qualified Policy Server host name in the Server field.
   Consider the following items:

   - You can enter a valid IPv4 address or IPv6 address.

   - If you do not specify a port, the registration defaults to 44442, which is the default Policy Server authentication port.

   The Administrative UI opens and is registered with the Policy Server.

The Administrative UI is now installed and registered.

# Configure an External Administrator Store for UI Administrators (Optional)

The policy store is the default repository for administrator identities. After you install and configure the Administrative UI, we recommend that you configure an external administrator store for UI administrators. You can use an LDAP directory server or a relational database as an external administrator store. For details, see Configure an External Administrator Store (https://docops.ca.com/display/sm1252sp1/Configure+an+External+Administrator+Store).

# (Optional) Install and Configure Additional Administrative UIs for High Availability

Install more than one Administrative UI to be sure that unexpected outages do not prevent you from managing CA Single Sign-On objects. Consider the following before installing another Administrative UI:

- Register the Administrative UI with a Policy Server that is not already sharing a trusted connection with an Administrative UI. Registering with another Policy Server prevents a single Policy Server outage from disabling both GUIs.
- An Administrative UI cannot failover to multiple Policy Servers. However, you can configure the Administrative UI to manage multiple Policy Servers.
- If the existing Administrative UI is configured for external CA Single Sign-On administrator authentication:
  - Configure the new Administrative UI with the same external store. Configuring the new Administrative UI with the same external store helps ensure that all CA Single Sign-On administrators are available to all GUIs.
  - Be sure to configure subsequent Administrative UI connections to the same external store using the same network identifier. Mixing network identifiers for multiple Administrative UI connections to the same external store is not supported.
    **Example:** If you configured the first connection with 172.16.0.0, create subsequent connections with 172.16.0.0. If you configured the first connection with comp001@example.com (mailto:comp001@example.com), create subsequent connections with comp001@example.com (mailto:comp001@example.com).

# (Optional) Configure Additional Policy Server Connections for the Administrative UI

By default, the Administrative UI is configured with a single Policy Server. You can configure additional Policy Server connections and can administer these servers from the Administrative UI. For example, you can create connections to manage Policy Servers in development and staging environments.

For the Administrative UI to connect to multiple Policy Servers, use an external administrator store. An external user store is a requirement for extra Policy Server connections. Create the administrator accounts for the administrator identities in the store. The accounts enable the Administrative UI to locate administrator records in the external store.

**Follow these steps:**

1. Configure a connection from the Administrative UI to an external administrator user store.

   **Note**: If the Administrative UI is using the policy store as its source of administrator identities, you cannot configure extra Policy Server connections.

2. Run the Administrative UI registration tool.

3. Gather the registration information.

4. Configure the connection to Policy Server.

5. (Optional) Modify the default Policy Server connection.

# Run the Administrative UI Registration Tool

You run the Administrative UI registration tool to create a client name and passphrase. A client name and passphrase pairing are values that the Policy Server uses to identify the Administrative UI you are registering. You submit the client and passphrase values from the Administrative UI to complete the registration process.

**Follow these steps:**

1. Open a command prompt from the Policy Server host system.

2. Run the following command:

   ```
   XPSRegClient client_name[:passphrase] -adminui -t timeout -r retries -c comment
   -cp -l log_path -e error_path-vT -vI -vW -vE -vF
   ```

   - **client_name**
     Identifies the Administrative UI that is being registered. Enter a unique value.

- **passphrase**
  Defines the password that is required to complete the registration of the Administrative UI. The passphrase has the following requirements:

  - The passphrase must contain at least six (6) characters.
  - The passphrase cannot include an ampersand (&) or an asterisk (*).
  - If the passphrase contains a space, it must be enclosed in quotation marks.
  - If you are registering the Administrative UI as part of an upgrade, you can reuse a previous passphrase.

  **Note**: If you do not specify the passphrase in this step, XPSRegClient prompts you to enter and confirm one.

- **-adminui**
  Defines that an Administrative UI is being registered.

- **(Optional) -t *timeout***
  Defines the time period in minutes in which you must log in to the Administrative UI from the time you install and register it with Policy Server. Policy Server denies the registration request if the time period expires.
  **Default:** 240 (four hours)
  **Minimum:** 1
  **Maximum:** 1440 (one day)

- **(Optional) -r *retries***
  Specifies how many failed attempts are allowed when you register the Administrative UI. A failed attempt can result from submitting incorrect administrator credentials when logging in to the Administrative UI for the first time.
  **Default:** 1
  **Maximum:** 5

- **(Optional) -c *comment***
  Inserts the specified comments into the registration log file for informational purposes. Surround comments with quotes.

- **(Optional) -cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The registration tool prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes.

- **(Optional) -l log_path**
  Specifies where to export the registration log file.
  **Default:***siteminder_home*\log

- **(Optional) -e error_path**
  Sends exceptions to the specified path.
  **Default:** stderr

- **(Optional) -vT**
  Sets the verbosity level to TRACE.
- **(Optional) -vI**
  Sets the verbosity level to INFO.
- **(Optional) -vW**
  Sets the verbosity level to WARNING.
- **(Optional) -vE**
  Sets the verbosity level to ERROR.
- **(Optional) -vF**
  Sets the verbosity level to FATAL.

The registration tool lists the name of the registration log file and prompts for a passphrase.

3. Press Enter.
   The registration tool creates the client name and passphrase pairing.

You can now register the Administrative UI with a Policy Server. You complete the registration process from the Administrative UI.

# Gather Registration Information

The Administrative UI requires the following information to register it with Policy Server:

- **Client name**—The client name that you specified using the XPSRegClient tool.
- **Passphrase**—The passphrase that you specified using the XPSRegClient tool.
- **Policy Server host**—The IP address or name of the Policy Server host system.
- **Policy Server authentication port**—The port on which the Policy Server is listening for authentication requests.
  **Default:** 44442

# Configure the Connection to Policy Server

Configure connections to Policy Server to manage CA Single Sign-On objects.

**Follow these steps**:

1. Log into the Administrative UI with an account that has super user permissions.
2. Click **Administration**, **Admin UI**.
3. Click **Policy Server Connections**, **Register Policy Server Connection**.

4. Type a connection name in **Name**.

5. Type the Policy Server host name or IP address in **Policy Server Host**.

6. Type the Policy Server authentication port in **Policy Server Port**.
   **Note:** This value must match the value in the Authentication port (TCP) field on the Settings tab in the Policy Server Management Console. The default authentication port is 44442. To determine the port number, open the Settings tab in the Policy Server Management Console.

7. Type the client name and passphrase you created using the registration tool in the respective fields.

8. Select a FIPS mode. If you installed the Policy Server in FIPS-compatibility mode, select **Compatibility** mode. If you installed the Policy Server in FIPS-only mode, select **FIPS only** mode.

9. Click Submit.
   The connection between the Administrative UI and Policy Server is configured.
   The Administrative UI login screen contains a list of Policy Servers to which the Administrative UI is registered. Policy Server that was registered first is the default connection.

## (Optional) Modify the Default Policy Server Connection

The Administrative UI login screen contains a list of Policy Servers to which the Administrative UI is registered. By default, the Policy Server that was registered first appears as the default connection. You can modify the list to have another Policy Server connection appear as the default.

**Follow these steps:**

1. Click Administration, Admin UI.

2. Click Policy Server Connections, Modify Policy Server Connection.

3. Specify search criteria and click Search.
   Administrative UI connections matching the criteria appear.

4. Select the connection you want and click Select.

5. Click the arrow icon in the Advanced group box.

6. Select the Default Connection check box and click Submit.
   The Policy Server connection is configured as the default connection.

# Re-register Administrative UI

This section explains the various approaches to re-register the Administrative UI with Policy Server.

# With Policy Server Shut Down Method

We recommend you to follow this method to re-register Administrative UI.

**Follow these steps:**

1. Stop the Administrative UI service.
2. Stop the Policy Server.
3. From the Administrative UI installation directory, navigate to Server\Default.
4. Delete the folders: data, log, tmp, and work.
5. Run the XPSRegClient command on the Policy Server machine.
   **Example:**

   ```
   XPSRegClient siteminder:<password> -adminui-setup -vT
   ```

6. Start the Policy Server.
7. Start the Administrative UI service.
8. Access the Administrative UI web page to complete the registration.
   **Example:**

   ```
   http://<Admin UI hostname>:8080/iam/siteminder/adminui
   ```

# Without Policy Server Shut Down

1. Stop the Administrative service or shut down the JBoss application server.
2. From the Administrative UI installation directory, navigate to Server\Default.
3. Delete the folders- data, log, tmp, and work.
4. Using the XPSExplorer utility, remove the trusted host that is created by XPSRegClient.
   For information about deleting the trusted host, see the section **XPSExplorer-delete Trustedhost** below.
5. Using the XPSSecurity utility, remove the Administrative UI directory.
   For information about deleting the Administrative UI directory, see the section **XPSSecurity-delete Administrative UI Directory** below.
6. Run the XPSRegClient command on the Policy Server machine.

   ```
   XPSRegClient siteminder:<password> -adminui-setup -vT
   ```

7. Start the Administrative UI service.
8. Access the Administrative UI web page to complete the registration.

```
http://<hostname>:8080/iam/siteminder/adminui
```

## Shut down JBOSS

1. From the SiteMinder home directory, navigate to siteminder/adminui/bin.
2. Run one of the following files to shut down the JBoss server:

- Windows- shutdown.bat
- UNIX- shutdown.sh

## XPSExplorer-delete Trustedhost

**To delete the TrustedHost**

1. Open the command prompt on the Policy Server.
2. Navigate to the <Policy Server installation directoryr>/bin folder.
3. Run the XPSExplorer utility.
4. Enter the number that corresponds to the Trusted Host.
5. Type 's' to search for objects and hit Enter key.
6. Locate the Trusted Host Object with the Description as "Generated by XPSRegClient" and Name as the Administrative UI Host name.
   **Example:**
   3-CA.SM::TrustedHost@24-xpsagent-fwrk-1cc9-991a-062X4CC9A2EB
   a. Name : "AdminUI Host name"
   b. Desc : "Generated by XPSRegClient"
   c. IpAddr : "0000:0000:0000:0000:0000:FFFF:"
   d. RolloverEnabled : false
7. Confirm the Object ID for the Trusted Host Object with the Name and Desc that corresponds to the Administrative UI Host.
   The Object ID prefaces CA.SM::TrustedHost@24-xpsagent-fwrk~ . In the example above, the Object ID is 3.
8. Type the Object ID number which corresponds to the Administrative UI Trusted Host object, and press the Enter key.
9. Type 'd' to delete the selected Object ID, and press the Enter key.
10. Repeat the command 'q' to navigate to the MAIN MENU.
11. Type 'q' to exit the XPSExplorer.

## XPSSecurity-delete Administrative UI Directory

You can delete the Administrative UI directory using the XPSSecurity utility.

Verify whether the XPSSecurity utility is in the directory siteminder_home/bin, where siteminder_home specifies the Policy Server installation path.

**Note:** If XPSSecurity is not present, download the Policy Server installation zip from the CA Support site. The tool is in the zip file.

**To delete the Administrative UI Directory**

1. Open a command window and run the following command:

   ```
   XPSSecurity
   ```

   The main menu appears.

2. Enter **A.**

   The menu lists the administrators. Each administrator is prefixed with a number. Search for SM_ADMIN-DIRECTORY.

   **For example:**

   7 - SiteMinder Administrative UI Directory User

   SM-ADMIN-DIRECTORY

   Used by the Administrative UI for authenticating administrators.

3. Enter **D.**
   The Administrative UI directory is deleted.

## With Policy Server Shut Down Method on WebLogic Admin Server

**Follow these steps:**

1. Run the following command to stop the WebLogic server:

   ```
   <weblogic_path_to_domain>\bin\stopWebLogic.sh
   ```

2. Stop the Policy Server.

3. From the WebLogic domain path, delete the "data" folder.

4. Run the XPSRegClient command on the Policy Server machine.

   ```
   XPSRegClient siteminder:<password> -adminui-setup -vT
   ```

5. Start the Policy Server.

6. Start the WebLogic server.
7. Access the Administrative UI web page to complete the registration.

   ```
   http://<hostname>:<port>/iam/siteminder/adminui
   ```

# Uninstall the Administrative UI from a Windows System (stand-alone)

Uninstall the Administrative UI when it is no longer required on the system.

**Follow these steps::**

1. Stop the application server using the following steps:
   a. From the Administrative UI host, open the Windows Services console.
   b. Stop the Administrative UI service.
2. Open the Windows Control Panel and go to the list of programs.
3. Right-click CA Single Sign-On Administrative UI.
4. Click Uninstall/Change.
5. Follow the instructions of the wizard.

   **Note:** If you are prompted to remove a shared file, click No to All.

6. If requested, reboot the system.
7. Open the Windows Control Panel and go to the list of programs.
8. Right-click Administrative UI Prerequisite Installer.
9. Follow the instructions of the wizard.

   **Note:** If you are prompted to remove a shared file, click No to All.

10. If requested, reboot the system.
    The Administrative UI and the required third–party components are uninstalled.

# Uninstall the Administrative UI from a UNIX System (stand-alone)

Uninstall the Administrative UI when it is no longer required on the system.

**Note**: Do not manually remove the installation directories to uninstall this component. Execute the uninstall shell script. If you only remove the installation directories, related registries can be left behind. If you try to re–install this component on this host system, the entries can prevent a successful installation.

**Follow these steps::**

1. Stop the application server using the following steps:

    a. Navigate to *install_home*/CA/siteminder/adminui/bin
    *administrative_ui_install*
    Specifies the Administrative UI installation path.

    b. Type shutdown.sh and press Enter

2. Open a shell and navigate to:

    ```
    administrative_ui_home/CA/SiteMinder/adminui/install_config_info
    ```

    *administrative_ui_home*
    Specifies the Administrative UI installation path.

3. Run the following command:

    ```
    ./sm-wamui-uninstall.sh
    ```

    The process to remove the Administrative UI starts.

4. Follow the prompts to uninstall the Administrative UI.
    The installer prompts you when the Administrative UI is uninstalled.

5. Open a command window and navigate to:

    ```
    administrative_ui_home/CA/SiteMinder/webadmin/install_config_info
    ```

    *administrative_ui_home*
    Specifies the Administrative UI installation path.

6. Run the following command:

    ```
    ./prerequisite-uninstall.sh
    ```

7. Follow the prompts to uninstall the Administrative UI prerequisite components.
    The installer prompts you when the third–party components are uninstalled.
    The Administrative UI and the required third-party components are uninstalled.

# Uninstall the Administrative UI from a JBoss or WebLogic Server (Windows)

Uninstall the Administrative UI from an existing JBoss Application Server when it is no longer required on the system.

**Follow these steps:**

1. Stop the application server:
    a. From the Administrative UI host system, open the Start Task Engine Command prompt.
    b. Enter the following keyboard combination:
    `Ctrl+c`
2. Open the Windows Control Panel and go to the list of programs.
3. Right–click CA Single Sign-On Administrative UI.
4. Click Uninstall/Change.
5. Follow the instructions of the wizard.

    **Note:** If you are prompted to remove a shared file, click No to All.

6. If requested, reboot the system.
    The Administrative UI is uninstalled.

# Uninstall the Administrative UI from an Existing JBoss or WebLogic UNIX System

Uninstall the Administrative UI from an application server when it is no longer required on the system.

**Note**: Do not manually remove the installation directories to uninstall this component. Execute the uninstall shell script. If you only remove the installation directories, related registries can be left behind. If you try to re–install this component on this host system, the entries can prevent a successful installation.

**Follow these steps:**

1. Stop the application server:

    a. From the Administrative UI host system, open the Start Task Engine Command prompt.

    b. Enter the following keyboard combination:

    ```
    Ctrl+c
    ```

2. Open a shell and navigate to the following directory:

    *administrative_ui_home*/CA/adminui/install_config_info

    *administrative_ui_home* specifies the Administrative UI installation path.

3. Run the following command:

    ```
    ./smwam-ui-uninstall.sh
    ```

    The process to uninstall the Administrative UI starts.

4. Follow the prompts to uninstall the Administrative UI.

The installer informs you when the Administrative UI is removed.

# Uninstall the Administrative UI from a WebSphere Windows System

Uninstall the Administrative UI when it is no longer required on the system.

**Follow these steps:**

1. Stop WebSphere using the following steps:

    a. From a command prompt, navigate to *profile*\bin
    *profile*
    Specifies the path of the WebSphere profile name you created for the Administrative UI
    **Example:** C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSvr01\bin

    b. Type stopServer.bat*identifier* and press Enter.
    *identifier*
    Specifies the identifier for the WebSphere installation.
    **Example:** stopServer.bat Server1

2. Open the Windows Control Panel and go to the list of programs.

3. Right–click CA CA Single Sign-On Administrative UI.

4. Click Uninstall/Change.

5. Follow the instructions of the wizard.

   **Note:** If you are prompted to remove a shared file, click No to All.

6. If requested, reboot the system.
   The Administrative UI is uninstalled.

# Uninstall the Administrative UI from an Existing WebSphere UNIX System

Uninstall the Administrative UI from an existing application server when it is no longer required on the system.

**Note**: Do not manually remove the installation directories to uninstall this component. Execute the uninstall shell script. If you only remove the installation directories, related registries can be left behind. If you try to re–install this component on this host system, the entries can prevent a successful installation.

**Follow these steps:**

1. Stop the application server.

   a. From a command prompt, navigate to *profile*\bin
   *profile*
   Specifies the path of the WebSphere profile name you created for the Administrative UI

   b. Enter the following command:

   ```
   stopServer.shidentifier
   ```

   *identifier*
   Specifies the identifier for the WebSphere installation.
   **Example:** stopServer.bat Server1

2. Open a shell and navigate to the following directory:

   ```
   administrative_ui_home/CA/adminui/install_config_info
   ```

   *administrative_ui_home*
   Specifies the Administrative UI installation path.

3. Run the following command:

   ```
   ./smwam-ui-uninstall.sh
   ```

   The process to uninstall the Administrative UI starts.

4. Follow the prompts to uninstall the Administrative UI.
   The installer informs you when the Administrative UI is removed.
   The Administrative UI is uninstalled.

# Install CA SiteMinder® SPS

You can install one or more instances of CA Access Gateway on the same computer. If the installation is successful, the installer installs the Secure Proxy Configuration Wizard.



# Verify Prerequisites

Before you install or upgrade, verify the following prerequisites:

- CA Access Gateway must not be installed on a system where Policy Server is installed.
- Ensure that Policy Server is running.
- Open port 7680 between CA Access Gateway and Policy Server.
- If you are installing CA Access Gateway on Linux, complete the following steps:
  - Ensure that the user used for installing CA Access Gateway has write permissions on the /opt directory.
  - The folder where you install CA Access Gateway must have sufficient permissions (755).
  - CA Access Gateway runs as the **nobody** user on UNIX. If you prefer not to run CA Access Gateway as this user, create an alternate user and assign the necessary permissions.
- If you are installing CA Access Gateway on RHEL, verify that you installed the following packages:
  Note: We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

  The following list describes the commands to install the required libraries on the host system:
  **RHEL 5.x**
  yum install -y ncurses-libs.i686

  **RHEL 6.x**

- yum install -y ncurses-libs.i686
- yum install keyutils-libs.i686

- If you are installing CA Access Gateway on a RHEL 5.x or 6.x (64-bit) system, verify that you installed the following libraries:
  - yum install -y libstdc++.i686
  - yum install -y libexpat.so.0
  - yum install -y libuuid.i686
- If you are installing CA Access Gateway on an RHEL 5.5 computer, verify that you installed the Legacy Software Development package on the computer.
- JCE patches required -- The current Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction patches are required to use the Java cryptographic algorithms. To locate the JCE package for your operating platform, go to the Oracle website.

  Apply the patches to the following files on your system:
  - local_policy.jar
  - US_export_policy.jar

  These files are in the following directories:
  **Windows:***jre_home*\lib\security
  **UNIX:***jre_home*/lib/security

  *jre_home* specifies the location of the Java Runtime Environment installation.
- Increase the source of randomness for the entropy pool. Use one of the following options:
  - (Most secure) Install a *hardware entropy generator* and configure the rngd daemon to populate /dev/random by entering the following command:

    ```
    rngd -r /dev/device_name -o /dev/random -b
    ```

    *device_name* is character device in use. The device name varies depending on the hardware random number generator that you are using, for example, /dev/hwrng.

    For more information about the rngd daemon, see the RedHat documentation.
  - (Good security) Configure the rngd daemon to populate /dev/random by entering the following command:

    ```
    rngd -r /dev/urandom -o /dev/random -b
    ```

    Third-party alternatives to the rngd entropy daemon are also available.
  - (Least secure) Configure a symbolic link between /dev/urandom and /dev/random by entering the following commands:

```
mv /dev/random /dev/random.org (http://random.org)ln -s /dev/urandom /dev/random
```

- Ensure that the CA RiskMinder service is running. To check the status, perform the following steps:

  **Windows**

  1. a. Open the Task Manager and verify that the arrfserver process is running.

     b. Navigate to *policy_server_installation_path*\aas\logs.

     c. Open the cariskminderstartup.log file and verify that the following line exists at the end of the file:

*CA RiskMinder Service READY*

**UNIX**

  1. a. Run the ps command and verify that the arrfserver and arrfwatchdog processes are running.

     b. Navigate to *policy_server_installation_path*/aas/logs.

     c. Open the cariskminderstartup.log file and verify that the following line exists at the end of the file:

*CA RiskMinder Service READY*

# Install CA Access Gateway

You can install CA Access Gateway on Windows or UNIX. CA Access Gateway sets the instance name of the first installation as **default**. You cannot modify the default value or cannot use the same name for any other instance.

## Install on Windows

**Follow these steps:**

1. Download the installer from CA Support.
2. Double-click **ca-proxy-<*version*>-<*operating_system*>.exe**.
3. Review the prerequisites that are required for proceeding with the installation.
4. Click Next when you are ready.
5. Accept the license agreement and click Next.
6. Specify the installation location and click Next.

7. Select the Java binary that is in the bin folder of the JDK installation.
   For example: C:\Program Files\Java\jdk1.8.0_51\bin\java.exe

8. Click Next.

9. Review the installation summary and click Install.

10. Click Done when the installation is complete.

## Install on UNIX

**Follow these steps:**

1. Download the following installer from CA Support:

   ```
   Solaris: ca-proxy-12.5-sol.bin
   Linux: ca-proxy-12.5-rhel30.bin
   ```

2. Execute the following command to initiate the installer:

   ```
   Solaris: sh ca-proxy-12.5-sol.bin
   Linux: sh ca-proxy-12.5-rhel30.bin
   ```

3. Review the installation requirements and press Enter to continue.

4. Follow the screen prompt to read the license agreement.

5. Type **Y** when prompted to accept the license agreement and press Enter.

6. Specify the installation location and press Enter.

7. Type the number corresponding to the Java binary that is in the bin folder of the JDK installation, and press Enter.

8. Review the install summary and press Enter.

9. Exit the installer when the installation is complete.

You can check the InstallLog file to verify that the installation is successful.
**Default Location**: *sps_home*\install_config_info\CA_SiteMinder_Secure_Proxy_Server_InstallLog

# Install Multiple Instances of CA Access Gateway

You can install multiple CA Access Gateway instances on the same computer. Each instance uses a unique instance name and port for communication, and creates a separate directory structure and services.

## Install Multiple Instances on Windows

**Follow these steps:**

1. Navigate to the location where you downloaded the installer.
2. Double-click **ca-proxy-<*version*>-<*operating_system*>.exe**.
3. Review the installation requirements and click Next.
4. Accept the license agreement and click Next.
5. Choose **New instance** as the install type.
6. Review the criteria to name an instance and enter a name for the new instance.
7. Click Next.
8. Specify the installation location and click Next.
9. Select the Java binary that is in the bin folder of the JDK installation.
   For example: C:\Program Files\Java\jdk1.8.0_51\bin\java.exe
10. Click Next.
11. Review the installation summary and click Install.
12. Click Done when the installation is complete.
13. (Optional) To install more instances, perform Steps 2-12 on the same computer.

# Install Multiple Instances on UNIX

**Follow these steps**:

1. Navigate to the location where you downloaded the installer.
2. Execute the following command to initiate the installer:

   ```
   Solaris: sh ca-proxy-12.5-sol.bin
   Linux: sh ca-proxy-12.5-rhel30.bin
   ```

3. Review the installation requirements and press Enter to continue.
4. Follow the screen prompt to read the license agreement.
5. Type **Y** when prompted to accept the license agreement and press Enter.
6. Type **1** to install a new instance.
7. Review the criteria to name an instance, enter a name for the new instance, and press Enter.
8. Specify the installation location and press Enter.
9. Choose the Java binary that is in the bin folder of the JDK installation. Type the number and press Enter.

10. Review the install summary and press Enter.
11. Exit the installer when the installation is complete.
12. (Optional) To install more instances, perform Steps 2-11 on the same computer.

Proceed with the configuration of each instance.

# Reinstall CA Access Gateway

You can reinstall CA Access Gateway to troubleshoot configuration issues.

## Reinstall on Windows

**Follow these steps**:

1. Navigate to the location where you downloaded the installer.
2. Double-click **ca-proxy-version-win64.exe**.
3. Review the installation requirements and click Next.
4. Accept the license agreement and click Next.
5. Choose **View existing instances** and click Next.
   A list of instances that are installed on the computer is displayed.
6. Select the instance and click Next.
   CA Access Gateway verifies if the selected instance can be reinstalled or upgraded, and displays a message accordingly.
7. If the selected instance can be reinstalled, click OK.

## Reinstall on UNIX

**Follow these steps**:

1. Navigate to the location where you downloaded the installer.
2. Execute the following command to initiate the installer:

   ```
   Solaris: sh ca-proxy-12.6-sol-64.bin
   Linux: sh ca-proxy-12.6-rhas64.bin
   ```

3. Review the installation requirements and press Enter to continue.
4. Follow the screen prompt to read the license agreement.

5. Type **Y** when prompted to accept the license agreement and press Enter.

6. Type **2** and press Enter.
   A list of instances that are installed on the computer is displayed.

7. Select the instance and click Next.
   CA Access Gateway verifies if the selected instance can be reinstalled or upgraded, and displays a message accordingly.

8. If the selected instance can be reinstalled, press Enter.

# Uninstall CA Access Gateway

To uninstall from Windows, perform the following steps:

1. Open the command prompt and navigate to the root installation directory.

2. Execute the following command for each instance you want to uninstall:

   ```
   ca-sps-uninstall.cmd
   ```

To uninstall from UNIX, perform the following steps:

1. Open a console window and navigate to the root installation directory.

2. Execute the following command to source the CA Access Gateway environment:

   ```
   source ca_sps_env.sh
   ```

3. Run the following program:

   ```
   ./ca-sps-uninstall.sh
   ```

**Note**: If you have modified any files such as server.conf, the uninstall program does not remove these files or their parent folders automatically. You must delete the files and folders manually.

# SDK

The following sections detail how to install the SDK.

# SDK System Requirements

## Operating Systems

To learn about operating system support for the SDK, see Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM). The SDK platform support matrix is typically included with this matrix.

## JRE Requirement

Verify that you have the required JRE version installed. For the required version, refer to Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

Applications developed with the current release of the SDK cannot be run against Policy Server versions prior to version 6.0. Applications developed with a version of the SDK that worked with the Policy Server v5.0, v5.5, and 6.x, will continue to work with the current release of the Policy Server.

# Considerations Before Installing the SDK

**Contents**



## System Locale Must Match the Language of Installation and Configuration Directories

To install and configure a CA Single Sign-On component to a non-English directory, set the system to the same locale as the directory. Also, make sure that you installed the required language packages so the system can display and users can type localized characters in the installer screens.

For the details on how to set locale and required language packages, refer to respective operating system documents.

# Considerations for Localized Installations

When installing the CA Single Sign-on SDK on a system with a non-English operating system, remember that the smpolicyapi is an UTF-8 based API. The library expects UTF-8 encoded strings as input. It returns UTF-8 encoded strings as output parameters.

# Installation of ETPKI Libraries

The CA Single Sign-On SDK installer ships the ETPKI installer does not install the ETPKI libraries, which can lead to errors when you run the SDK sample applications. You must install the ETPKI libraries separately when installing the SDK on a system without other CA Single Sign-On components.

**Default Location of 32-bit ETPKI**: /etpki-install

**Default Location of 64-bit ETPKI**: /etpki-install-64

The ETPKI r4.x (CAPKI) installer runs in silent mode without any user interaction. The ETPKI installer has the following usage:

```
setup.exe {install|remove} caller=callerID [options...]
```

- callerID

  Specifies the parent application that is installing ETPKI r4.x (CAPKI). This identifier is user-defined; it specifies the parent product. When multiple subcomponents of a product rely on CAPKI, each component uses a different identifier.

  **Limits**: 255 characters

### Options

- instdir=*user_supplied_directory*

  Specifies the initial installation directory. The installer considers this option only when these libraries are the first CA shared component on this system.

- verbose/veryverbose

  Enables diagnostic output with more or less detail.

### UNIX only

- env={none|user|all}

  Specifies whether you can set environment variables for the specified user as follows:

- none — No environment variables set (default)
  - user — The current user only (SHOME/.profile)
  - all — all users (/etc/profile)

  Using the env=user or env=all option, the ETPKI installer creates the following environment variables:

- CASHCOMP — Points to the parent directory of the CAPKI installation
  - CALIB — Points to the $CASHCOMP/lib directory
  - CABIN — Points to the $CACHCOMP/bin

**Example**

```
[root@talon etpki-install-64]# ./setup install caller=01010101 instdir=/home/CA
/etpki verbose env=user
```

For more information, see the readme.txt file included with the SDK installation in the etpki-install subfolder.

# Install the SDK on UNIX in Console Mode

No special accounts or privileges are required to install the CA Single Sign-On SDK. Instructions for installing a first version of the SDK and upgrading from an existing version are the same.

Do not install the SDK in the installation path with the Policy Server or Web Agent. The SDK can possibly have different versions of the same support libraries.

On UNIX, the installation executable file is ca-sdk-12.52sp1-*platform*.bin.

You can install the SDK in GUI mode or console mode.

**To Install the SDK in UNIX Console Mode**

1. Close all programs.
2. Download the CA Single Sign-On SDK from the CA Technical Support site (http://www.ca.com/support).
3. In a UNIX shell, navigate to the directory that corresponds to your platform (solaris, aix, linux, or hpux).
4. Enter the following command:

   ```
   sh ./ca-sdk-12.52sp1-platform.bin -i console
   ```

   - platform

     Replace *platform* with sol, aix, linux, suse, or hp.

For example, on Solaris platforms, the command is:

```
sh ./ca-sdk-12.52sp1-sol.bin -i console
```

Follow the wizard.

# Install the SDK on UNIX in GUI mode

No special accounts or privileges are required to install the CA Single Sign-On SDK. Instructions for installing a first version of the SDK and upgrading from an existing version are the same.

Do not install the CA Single Sign-On SDK in the same path as the Policy Server or Web Agent. The SDK can possibly have different versions of the same support libraries.

On UNIX, the installation executable file is ca-sdk-12.52sp1-*platform*.bin.

You can install the SDK in GUI mode or console mode.

**To Install the SDK in UNIX GUI Mode**

1. Close all programs.
2. Download the SDK from the CA Technical Support site (http://www.ca.com/support).
3. In a UNIX shell, navigate to the directory that corresponds to your platform (solaris, aix, linux, or hpux).
4. Enter the following command:

   ```
   sh ./ca-sdk-12.52sp1-OS.bin
   ```

   - OS

     Replace *OS* with sol, aix, linux, suse, or hp.

   For example, on Solaris platforms, the command is:

   ```
   sh ./ca-sdk-12.52sp1-sol.bin
   ```

5. Follow the wizard.

# Install the SDK on Windows

No special accounts or privileges are required to install the CA Single Sign-On SDK. Instructions for installing a first version of the SDK and upgrading from an existing version are the same.

Do not install the CA Single Sign-On SDK in the installation path with the Policy Server or Web Agent. The SDK can possibly have different versions of the same support libraries.

**To install the SDK**

1. Close all programs.
2. Download the SDK from the CA Technical Support site (http://www.ca.com/support).
3. Navigate to the win32 directory and run the following program:

   ```
   ca-sdk-version-win32.exe
   ```

Follow the wizard.

# Windows Server 2008 System Considerations

For Windows Server 2008, the User Account Control feature helps prevent unauthorized changes to your system. When the User Account Control feature is enabled on the Windows Server 2008 operating environment, prerequisite steps are required before doing any of the following tasks with a CA Single Sign-on component:

- Installation
- Configuration
- Administration
- Upgrade

**Note:** For more information about which CA Single Sign-on components support Windows Server 2008, see the CA Single Sign-on Platform Support matrix.

**To run CA Single Sign-on installation or configuration wizards on a Windows Server 2008 system**

1. Right–click the executable and select Run as administrator.
   The User Account Control dialog appears and prompts you for permission.
2. Click Allow.
   The wizard starts.

**To access the CA Single Sign-on Policy Server Management Console on a Windows Server 2008 system**

1. Right–click the shortcut and select Run as administrator.
   The User Account Control dialog appears and prompts you for permission.
2. Click Allow.
   The Policy Server Management Console opens.

**To run CA Single Sign-on command–line tools or utilities on a Windows Server 2008 system**

1. Open your Control Panel.

2. Verify that your task bar and Start Menu Properties are set to Start menu and *not* Classic Start menu.

3. Click Start and type the following in the Start Search field:

   ```
   Cmd
   ```

4. Press Ctrl+Shift+Enter.
   The User Account Control dialog appears and prompts you for permission.

5. Click Continue.
   A command window with elevated privileges appears. The title bar text begins with Administrator:

6. Run the CA Single Sign-on command.

# Unattended Installation of the SDK on UNIX

After the CA Single Sign-On SDK has been manually installed, you can install it on the same system, or a different system, using the silent installation mode. An unattended installation uses one command that points to a properties file for installation preferences. The default properties template file (sdk-installer.properties in the install_config_info folder) can be modified to suit your requirements.

**To install the SDK in silent mode on UNIX**

1. Navigate to the directory where the SDK executable is located.

2. Enter the following command at a command prompt:

   ```
   sh ./ca-sdk-12.52sp1-platform.bin -f sdk-installer.properties -i silent
   ```

   - -f

     Specifies the name of the SDK installer properties file. If the properties file is not in the same directory as the installation executable file, specify the relative path to the properties file.

   - -i

     Specifies the installation mode.

   The installation is complete.

# Unattended Installation of the SDK on Windows

After the CA Single Sign-On SDK has been manually installed, you can install it on the same system, or a different system, using the silent installation mode. An unattended installation uses one command that points to a properties file for installation preferences. The default properties template file (sdk-installer.properties in the install_config_info folder) can be modified to suit your requirements.

**To install the SDK in silent mode on Windows**

1. Navigate to the directory where the SDK executable is located.
2. Enter the following command at a command prompt:

   ```
   ca-sdk-version-win32.exe -f sdk-installer.properties -i silent
   ```

   - -f

     Specifies the name of the SDK installer properties file. If the properties file is not in the same directory as the installation executable file, specify the relative path to the properties file.

   - -i

     Specifies the installation mode.

The installation is complete.

# Uninstallation of the SDK

**To uninstall the SDK from the UNIX console**

1. In a console window, navigate to the install_config_info/ca-sdk-uninstall directory within the SDK installation—for example:

   /export/ca/sdk/install_config_info/ca-sdk-uninstall

2. Run the following command:

   ```
   ./uninstall -i console
   ```

3. When prompted, press Enter to begin the uninstallation.
   When you are uninstalling the SDK in UNIX, make sure the JRE is in the PATH variable. If the JRE is not in the PATH variable, the following error occurs:

   ```
   No Java virtual machine could be found from your PATH environment variable. You
   must install a VM prior to running this program.
   ```

To set the JRE in your PATH, run the following two commands:

1. PATH=$PATH:*JRE_location*/bin

   For example: PATH=$PATH:/usr/bin/jdk141/jre/bin

2. export PATH

**To uninstall the SDK from Windows:**

1. From the Control Panel, double-click Add/Remove Programs.

2. Select CA Single Sign-on SDK *version* and click Change/Remove.

Follow the screen prompts, and click Close when done.

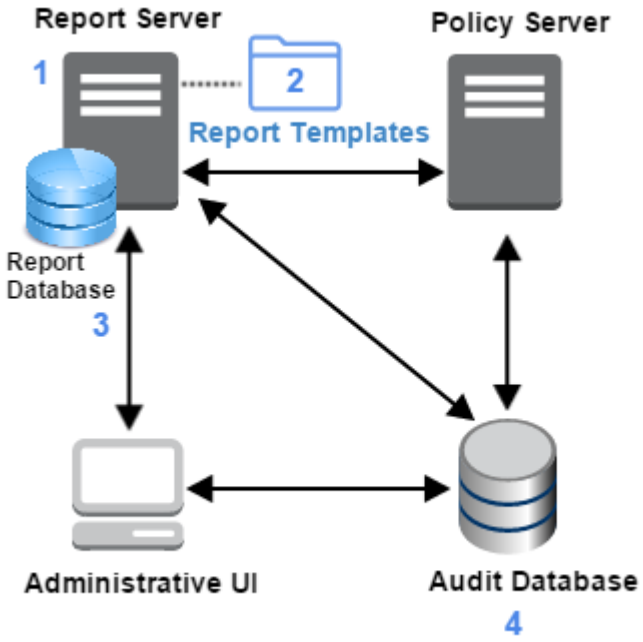
# Report Server and Reporting Databases

**Note:** Starting April 1, 2016, customers who are new licensees of (CA SSO) do not have rights to access or use the CA Report Server from CA Technologies as a component of the product. Customers who licensed CA SSO before April 1, 2016 continue to have rights to access the CA Report Server from CA Technologies and use the CA Report Server with CA SSO.

The CA Single Sign-On reporting feature requires that you install and configure the following components:

- A Report Server
- A report database
- An audit database

The following diagram shows a sample environment and lists the installation and configuration order of each component:




Complete the following steps:

1. **Install the Report Server**—Configure a report database during the installation.

2. **Install the Report Templates**—Run the Report Server Configuration Wizard to install the report templates. The wizard configures the Report Server to use a set of policy analysis and auditing report templates.

3. **Register the Report Server**—Register the report server by configuring a connection between:

   - The Report Server and a Policy Server
   - The Report Server and the Administrative UI

4. **Configure an audit database**—A separate audit database, which is registered with the Administrative UI, is required to run audit-based reports. The audit database manages policy analysis and audit-based reports.

The content in this section describes how to install and configure a Report Server, report database, and audit database. Use the Table of Contents to access the content.

# Report Server System Requirements

Adhere to the minimum system requirements for installing the Report Server and its associated components.

The following sections detail the minimum system requirements for installing the Report Server. For a list of supported CA and third-party components, refer to the Platform Support Matrix on the Technical Support site (https://support.ca.com/).


- Windows System Requirements for the Report Server (see page 489)
- UNIX System Requirements for the Report Server (see page 490)
- Browser Requirements (see page 491)
- Solaris and Red Hat Required Patch Clusters (see page 491)


# Windows System Requirements for the Report Server

Do not install the Report Server to a domain controller because is not a supported environment.

The Windows system where you install the Reports Server must meet the following minimum system requirements:

- **CPU**—Intel Pentium 4–class processor, 2.0 GHz.
- **Memory**—2 GB of RAM.
- **Available disk space**—10 GB.
  This requirement does not account for the disk space that is required to store reports.
- **Temp directory space**—1 GB.

- **Screen resolution**—1024 x 768 or higher resolution with 256 colors or better to view reports properly in the Administrative UI.
- **User account requirement:** Verify that y ou have access to a user account that is a member of the local Administrators group. The user account that runs the installer must be a member of the local Administrators group.

  **Important!** Do not install the Report Server to a system where the default Windows security settings for the local Administrator group have been modified.

- **Host name**—The Report Server host name cannot include any of the following characters:
  - An underscore (_)
  - A period (.)
  - A slash (/) (\)

# UNIX System Requirements for the Report Server

The UNIX system where you install the Reports Server must meet the following minimum system requirements:

- **CPU**
  - **Solaris:** SPARC v8plusSparc
  - **Red Hat Linux:** Intel Pentium 4–class processor, 2.0 GHz.
- **Memory**—2 GB of RAM.
- **Available disk space**—10 GB.
  This requirement does not account for the disk space that is required to store reports.
- **Temp directory space**—1 GB.
- **Screen resolution**—1024 x 768 or higher resolution with 256 colors or better to view reports properly in the Administrative UI.
- **Host IP**—Report Server host IP is configured properly in the /etc/hosts file to avoid an IP resolution problem.
- **User accounts**—You have access to a root user account and a non-root user account. You need a root-user account to start the installation and a non-root user account to complete the installation. The non-root user account Is the owner of a valid home directory and has write permissions to the home directory.

- **PATH environment variables**—The following commands and utilities must be installed on the operating system and available on the PATH environment variable for the root user account:

  ```
  awk, /bin/sh, chown, dirname, expr, grep, gzip, hostname, id, pwd, read, stty,
  sed, tail, tar, touch, ulimit, uname, which
  ```

  The PATH environment variable of the root–user account cannot inlcude GNU or third-party replacements for core system command–line tools or an individually downloaded and compiled version of the tool.

- **Locale**—At least one of the following variables is set to a valid utf8/UTF-8 locale:

  - LC_ALL
    Example: export LC_ALL=en_US.UTF-8
  - LANG
    Example: export LANG=en_US.UTF-8
  - LC_CTYPE
    Example: export LC_CTYPE=en_US.UTF-8

- **Host name**—The Report Server host name cannot include any of the following characters:

  - An underscore (_)
  - A period (.)
  - A slash (/) (\)

## Browser Requirements

Configure the browser that use to open the Administrative UI and launch reports to allow mixed content (HTTP/HTTPS). (The Administrative UI is HTTPS but calls to the Report Server are done using HTTP. For more information, see your browser documentation.)

## Solaris and Red Hat Required Patch Clusters

The Report Server requires specific Solaris and Red Hat patch clusters. Before you install the Report Server, update your Solaris or Red Hat system with the latest patches. If you do not install the required patches, the Report Server installation fails.

# Report and Audit Database Requirements

The Report Server requires a report database to run reports. The Report Server installer includes an embedded version of Sybase SAP SQL Anywhere that you can install as the report database.

Review the Report Server and report database requirements to ensure that to help ensure that your environment meets the minimum operating system and database requirements and that you complete the required prerequisite configuration.


- Microsoft SQL Server and Oracle Database Considerations (see page 492)
- Microsoft SQL Server as a Report and Audit Database (see page 493)
- Oracle as a Report and Audit Database (see page 493)
- Connectivity Requirements (see page 494)


If you do not use the embedded version of SQL Anywhere, use a supported version of one of the following databases:

**Windows**

- Microsoft SQL Server (SQL Server)
- Oracle

**Solaris and Linux**

- Oracle

# Microsoft SQL Server and Oracle Database Considerations

If using a Microsoft SQL Server or Oracle for the Report Server database, use the same database type for the audit store database. For example, if Oracle is the Report Server database, you cannot use Microsoft SQL Server for the audit store. However, if the embedded version of SQL Anywhere (recommended) is used for the Report Server database, Microsoft SQL Server or Oracle is supported as an audit store.

**Important!** The Report Server is a common component that CA products share. As such, the installer lets you configure database types and versions that CA Single Sign-On does not support but other CA products do. For a list of supported databases, see the Platform Support Matrix.

If you using any other database other than the embedded version of SQL Anywhere, verify the following requirements:

- The Database server host system has a fixed host name.
- You are using a supported database to function as the report database.
- A new, empty database is available.
- The database client and server are configured to use UTF-8 character encoding. For more information about the required settings for a Unicode configuration, see your vendor-specific documentation.

# Microsoft SQL Server as a Report and Audit Database

If you are using SQL Server as a report database, an audit database, or both, complete the following tasks on the Report Server host system:

- Verify that a supported SQL Server driver is installed on the Report Server host system.
- Create a data source name (DSN) that identifies each database. The Report Server uses the DSN to communicate with each database.
- Verify that the database is enabled for UTF-8 character encoding. For more information about the required settings for a Unicode configuration, see your SQL Server documentation.

# Oracle as a Report and Audit Database

If you are using Oracle as a report database, an audit database, or both, complete the following tasks on the Report Server host system:

- Verify that a supported Oracle Net client is installed.
- Verify that the database client and server are configured to use UTF-8 character encoding. For more information about the required settings for a Unicode configuration, see your vendor-specific documentation.
- Use an Oracle Net Service Name to identify each database in the tnsnames.ora file. The Report Server uses the service name to communicate with each database.
- Set the NLS_LANG variable to one of the following UTF-8 settings:
  - AMERICAN_AMERICA.WE8MSWIN1252
  - AMERICAN_AMERICA.AL32UTF8
- Set the ORACLE_HOME variable to *Oracle_Net_client*.
  *Oracle_Net_client* specifies the Oracle Net client installation path.
  **Windows example:** C:\oracle\product\11.1.0\client
  **UNIX example:** export ORACLE_HOME=/opt/oracle/product/11.1.0/client1
- Windows:
  Add the ORACLE_HOME variable to the system environment variables.
  **Example:** %Oracle_Home%\bin
- UNIX:
  Set the LD_LIBRARY_PATH variable to $ORACLE_HOME/lib32:$ORACLE_HOME/lib.
  **Example:** LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
- UNIX:
  Set the PATH variable to $ORACLE_HOME/bin:$PATH.
  **Example:** export PATH=$ORACLE_HOME/bin:$PATH

# Connectivity Requirements

If you use the embedded version of SQL Anywhere as a Report Server database, there are no database connectivity requirements. If you use Microsoft SQL Server or Oracle as a report server or audit database, a driver is required for connectivity.

Install the appropriate driver on the Report Server host system:

- Microsoft SQL Server—A supported Microsoft SQL Server driver.
- Oracle database—A supported Oracle Net client driver.

For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

# Install the Report Server

**Note:** Starting April 1, 2016, customers who are new licensees of CA Single Sign-On (CA SSO) do not have rights to access or use the CA Report Server from CA Technologies as a component of the product. Customers who licensed CA SSO before April 1, 2016 continue to have rights to access the CA Report Server from CA Technologies and use the CA Report Server with CA SSO.

The following information explains the prerequisites and procedure for installing the Report Server.

# Gather Information for the Installer

Review the following sections to identify the information required by the Report Server installer.

## Installation Credentials

Depending on the operating system where you install the Report Server, the installer requires one or more sets of credentials:

- **BusinessObjects Administrator password (Windows and UNIX)**
  The installer creates a default SAP BusinessObjects Enterprise administrator account (administrator). Use this account to import report templates and to access the BusinessObjects Content Management Console. Determine the password for this account.

  The password must meet the following composition criteria:

  - Include at least six characters.

- Cannot contain the word "administrator" in any form.
- Include at least two of the following character types:
  - Uppercase characters
  - Lowercase characters
  - Numerals
  - Punctuation

- **Non-root user account (UNIX)**—You need a root-user account to start the installation and a non–root user account to complete the installation. Identify the non–root user account:
  - User name
  - Group name

## SQL Anywhere Report Database

To use the embedded version of SQL Anywhere as the report database,collect the following information before you start the installation:

- **SQL Anywhere root password**
  The password for the SQLAnywhere root user account. You cannot change the name of the root user account. The installer defaults the name to root.

- **User**
  The name of the report database administrator account.
  Installer default: sa

- **Password**
  The password of the report database administrator account.

## Microsoft SQL Server Report Database

To use Microsoft SQL Server as the report database, collect the following information before you start the installation

- **Data Source Name (DSN)**
  The DSN that Report Server uses to communicate with the report database.

  The Report Server is compiled as 32-bit native binary and is designed to use 32-bit data source middleware connectivity. If you are installing to a Windows 64–bit operating system, be sure to create the DSN using odbcad32.exe. This executable is located in the following directory:

  *install_home*\Windows\SysWOW64

  *install_home* specifies the installation path of the Windows operating system.

- **Login ID**
  The name of the report database administrator account. This account must have the default (DBO) account permissions.

- **Password**
  The password of the report database administrator account.

## Oracle Report Database

To use Oracle as the report database, collect the following information before starting the installation:

- **Oracle Net Client Service name**—The service name that the Report Server must use to communicate with the report database.
  - Windows: The installer refers to the Oracle Net Client Service name as Server.
  - UNIX: The installer refers to the Oracle Net Client Service name as TNSNAME.
- **User name**—The name of the report database administrator account.
  The administrator account must have the following privileges:
  - create session
  - create table
  - create procedure

  You can use an administrator account with the CONNECT and RESOURCE roles enabled, but disable the Admin Option setting for both roles.
- **Password**
  The password of the report database administrator account.

## Apache Tomcat Installation

CA Single Sign-On only supports the Apache Tomcat server that is embedded with the Report Server. Gather the following information before starting the installation:

- **Connection port**
  The port to which Apache Tomcat is to connect and wait for requests.
  Default: 8080
- **Redirect port**
  Identify the port to which Apache Tomcat must redirect requests.
  Default: 8443
- **Shutdown port**
  The port to which the Apache Tomcat SHUTDOWN command must be issued.
  Default: 8005

## CABI Installation

Prior to installing the Report Server Configuration Wizard, ensure that the CABI installation creates the registry entry.

CABI Installation creates a registry entry in the following registry path:

**[HKEY_USERS\.DEFAULT\Software\SAP BusinessObjects\Enterprise\CMSClusterMembers]**

The registry entry has the Key Name and Value as shown:

- **Key Name:** @hostname:portnumber of CABI server
  **Example Key Name:** @cabihostname:6400

- **Value:** hostname:portnumber;
  **Example Value:** cabihostname:6400;

# Install the Report Server

**Important!** The Report Server is a common component that CA products share. As such, the installer lets you configure database types and versions that CA Single Sign-On does not support but other CA products do. For a list of supported databases, see the Platform Support Matrix.

Consider the following items *before* you install the Report Server:

- Do not install the Report Server on a host system with any other CA Single Sign-On component. That deployment is not supported. install the Report Server on a separate host system.

- Install the Report Server using the installation media on the Technical Support site. For a list of installation media names, see the *Policy Server Release Notes*.

- If you are installing to a Windows 64–bit operating system, be sure to create the DSN using odbcad32.exe. This executable is located in the following location:

  `install_home\Windows\SysWOW64`

*install_home* specifies the installation path of the Windows operating system.

- The Report Server installation includes the following components that run as processes:

  - The Content Management Server

  - The Server Intelligence Agent

  These components require TCP/IP ports to communicate. The installer lets you modify the default settings to prevent port conflicts on the Report Server host system.

- The installation zip contains multiple folders. The installer requires this folder structure. If you move the Report Server installer after extracting the zip, copy the entire folder structure to the same location. Execute the installation media from this folder structure.

## Install the Report Server on Windows Systems

**Follow these steps:**

1. Ensure that you have gathered the required information for the installer.
2. Exit all applications that are running.
3. See the CA Business Intelligence documentation to install CABI using *one* of the following databases:
   - Embedded SQL Anywhere.
   - Oracle as the existing database.
   - MySQL as the existing database.

   Choose a database type that is supported by the Report Server. For a list of supported database types and versions, see the Platform Support Matrix.

If applicable, after installing the report server, migrate data (https://wiki.ca.com/display/SITEMINDER/.Install+the+Report+Server+vRIO#id-.InstalltheReportServervRIO-MigrateDatafromCABI3.xto4.1SP3) from the previous release to the current release of CABI.

## Install the Reports Server on UNIX Systems

**Follow these steps:**

1. Ensure that you have gathered the required information for the installer.
2. Exit all applications that are running.
3. See the CA Business Intelligence documentation to install CABI using *one* of the following databases:
   - Embedded SQL Anywhere.
   - Oracle as the existing database.
   - MySQL as the existing database.

   Choose a database type that is supported by the Report Server. For a list of supported database types and versions, see the Platform Support Matrix.

   If applicable, after installing the report server, migrate data (https://wiki.ca.com/display/SITEMINDER/.Install+the+Report+Server+vRIO#id-.InstalltheReportServervRIO-MigrateDatafromCABI3.xto4.1SP3) from the previous release to the current release of CABI.

**Note:** After you install CABI 4.1 SP3, run the Post Install Utility (PostIntsall.sh) before you run the Reports Configuration Wizard. The Reports Configuration Wizard is dependent on the CASHCOMP variable that is configured by the Post Install Utility (PostIntsall.sh) of CABI 4.1 SP3.

# Reinstall the Report Server

To reinstall the Report Server on a system, first uninstall the existing instance. If you attempt to install over an existing instance, the installation fails.

# Troubleshoot the Report Server Installation

## Use Log Files to Aid Troubleshooting

Use the following files to troubleshoot the Report Server installation:

- **CA_Business_Intelligence_InstallLog.log**—Open this log first to view reported errors.
- **ca-install.log**—Scroll to the bottom of this file to view reported errors. Search for "BIEK_GetExitCode" to verify the returned value of the "BIEK_GetExitCode" function. If the returned value is not 0, then there is an installation error. Search for the following keywords to determine the cause of the error:
  - Error
  - Warning
  - CMS
  - InfoStore

The log files are located in a temporary location during the installation. The TEMP environment property on the system determines the temporary location. If the installation fails, you can locate the log file in this temporary location. After a successful installation, the log files are located at the top level of the Report Server installation directory.

## Resolve JBoss Port Conflicts

If JBoss is installed on the Report Server host system, port conflicts can occur. If you experience port conflicts after installing the Report Server, review the JBoss port information in the following files:

- **jboss-service.xml**
  Default location:*jboss_home*\server*server_configuration*\conf
- **server.xml**
  Default location: *jboss_home*\server*server_configuration*\deploy\jbossweb-tomcat55.sar
  Default value: default

  *jboss_home* specifies the JBoss installation path.
  *server_configuration* specifies the name of your server configuration.

> **Note:** If you change the jboss-service.xml or the server.xml file, restart the JBoss application server. For more information about these files, see the Red Hat documentation.

- **service-bindings.xml**
  This file is only present if the Administrative UI is installed on the Report Server host system using the stand-alone option. If you change this file, restart the Administrative UI application server.

  Default location: *administrative_ui_install*\webadmin\conf
  *administrative_ui_installation s*pecifies the Administrative UI installation path.

# Migrate Data from CABI 3.x to CABI 4.1 SP3

After installing the report server, you can migrate the data from a older version of CABI to the current version The data can be CA Single Sign-On-specific or the entire CABI data. Use the Upgrade Management Tool shipped with the CABI 4.1 SP3 installer to migrate the data.

> **Important!** Before you start the migration process, verify that the InputFileRepository service is running in CABI.

**Follow these steps:**

1. Launch the Upgrade Management Tool.
2. Select Complete Upgrade or Incremental Upgrade.
3. Do one of the following:
   - If you select Complete Upgrade, click Start.
   - If you select Incremental Upgrade, click Next.
4. Verify that the Upgrade Scenario has **Live to Live** selected.
5. Specify the CMS Name and the log in credentials of the administrator of the Source server and the Destination server. Click Next.
6. Select the objects that you want to migrate to the destination server and click Next. You can select the reports or the entire CABI data.
7. Click Start then click OK.

The selected objects are migrated to the destination server.

# ODBC Configuration on CABI Machine for Audit Reports

You can create crystal reports for CABI 4.x based on CABI 3.x.

**Follow these steps:**

1. Install both 32-bit and 64-bit Oracle clients of Administration type to create audit reports using CABI 4.x.

   **Note:** The ORACLE_HOME environment variable is set to 64-bit installation and in the PATH and LD_LIBRARY_PATH environment variables, 64-bit is followed by 32-bit.

2. Add the TNS entry in the oracle 64 bit TNSNAMES.ORA file. This file normally resides in the ORACLE HOME\NETWORK\ADMIN **directory.**

   **Note:** If you migrated CABI 3.x crystal reports, 32-bit installation should also include the same TNS entries as in 64-bit.

3. Test whether the added TNS name is pinging or not.

Example:

TNSName=
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST =<database server>)(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = <service name>)
)

The audit reports are now created using CABI 4.x.

# Install the Report Templates

Follow these instructions for installing Report templates.


- Gather Information for the Installer (see page 501)
- Install the Report Templates (see page 502)


# Gather Information for the Installer

Install the Report Templates by running the Report Server Configuration Wizard. The Report Server Configuration Wizard requires the following information:

- **BusinessObjects administrator password**—Identify the password for the default BusinessObjects administrator account. The Report Server installer creates a default administrative account during installation. A password for this account was required to complete the installation. The Report Server Configuration Wizard requires the password to use the default administrative account to install the report templates.

- **Audit database type**—Identify the type of database that is to function as an audit database. A separate audit database is required to run audit-based reports. The Report Server Configuration Wizard requires the database type to configure the Report Server to use a set of report templates that are based on the audit database type.
  You do not have to configure an audit database before running the Report Server Configuration Wizard.

# Install the Report Templates

The following sections detail how to install the report templates on Windows and UNIX.

## Before You Install the Report Templates

To install the report templates, run the Report Server Configuration Wizard using the following installation binaries or executables:

**Solaris and Linux:** ca-rs-config-version-*cr*-*os*.bin
*cr* is the change release version
*os* is **solaris** or **linux**

**Windows**: ca-rs-config-version-*cr*-win32.exe
*cr* is the change release version

For UNIX platforms, the installation binary requires executable permissions. Run the following command to add the permission:

```
chmod +x ca-rs-config-version-cr-os.bin
```

*cr* is the change release version
*os* is **solaris** or **linux**

**Important!** On UNIX system, install the report templates directly on the Report Server host system. Executing the Report Server Configuration Wizard across different subnets can cause it to crash.

## Install the Report Templates on Windows Systems

For a list of installation media names, see the *Policy Server Release Notes*.

**Follow these steps for an installation in Windows:**

1. Be sure that you have gathered the required information for the installer.
2. Exit all applications that are running.
3. Double-click **ca-rs-config-version-*cr*-win32.exe**.
   *cr* is the change release version
   The installer starts.
4. Follow the prompts.
5. Review the installation settings and click Install.
6. After the installation is complete, restart the Report Server.
   The Report Server is configured to use the report templates.

## Install the Report Templates on UNIX Systems (GUI Mode)

For a list of installation media names, see the *Policy Server Release Notes*.

**Follow these steps for a GUI Mode installation:**

1. Be sure that you have gathered the required information for the installer.
2. Exit all applications that are running.
3. Log in as the root user.
4. Open a shell and navigate to the installation binary.
5. Enter the following command:

   `./ca-rs-config-version-cr-os.bin`

   *cr* is the change release version
   *os* is **solaris** or **linux**

   The installer starts.
6. Follow each prompt and enter the required values.

   **Note**: Oracle is the only supported audit database for a Solaris Report Server. If you installed the Report Server on a Solaris system, you are not prompted for an audit database type. The Report Server Configuration Wizard automatically installs Oracle-specific report templates.
7. Review the installation settings and click Install.
8. Restart the Report Server.

The Report Server is configured to use the report templates.

## Install the Report Templates on UNIX Systems (Console Mode)

For a list of installation media names, see the *Policy Server Release Notes*.

**Follow these steps for a console mode installation:**

1. Be sure that you have gathered the required information for the installer.
2. Exit all applications that are running.
3. Log in as the root user.
4. Open a shell and navigate to the installation binary.
5. Enter the following command:
   **./ca-rs-config-version-*cr-os.*bin console**

   *cr* is the change release version
   *os* is **solaris** or **linux**

   The installer starts.
6. Follow the prompts and enter the required values.

   **Note**: Oracle is the only supported audit database for a Solaris Report Server. If you installed the Report Server on a Solaris system, you are not prompted for an audit database type. The Report Server Configuration Wizard automatically installs Oracle-specific report templates.

7. Review the installation settings and press Enter.
   The report templates are installed.
8. Restart the Report Server.

The Report Server is configured to use the report templates.

# Register the Report Server

Registering the Report Server requires access to the Policy Server host system, the Report Server host system, and the Administrative UI host system. The registration process:

- Establishes a trusted relationship between the Report Server and the Policy Server.
- Establishes a trusted relationship between the Report Server and the Administrative UI.
- Register the Report Server with the Policy Server (see page 507)

- Restart the Report Server (see page 508)
- Configure the Connection to the Administrative UI (see page 510)

### Create a Client Name and Passphrase

Run the XPSRegClient utility to create a client name and passphrase. The Policy Server uses a client name and passphrase to identify the Report Server you are registering. The XPSRegClient tool uses the client name and passphrase to register the Report Server with the Policy Server.

**To run the registration tool**

1. Open a command window on the Policy Server host system.

2. Navigate to *siteminder_home*/bin.
   *siteminder_home* specifies the Policy Server installation path.

3. Run the following command:

   ```
   XPSRegClient client_name[:passphrase] -report -t timeout -r retries
   -c comment -cp -l log_path -e error_path -vT -vI -vW -vE -vF
   ```

   - *client_name*
     Identifies the name of Report Server you are registering.
     **Value:** The value must be unique. For example, if you previously used reportserver1, enter reportserver2.
     Record this value. This value is required to complete registration process from the Report Server host system.

   - *passphrase*
     Specifies the password required to complete the Report Server registration.
     **Value:** The passphrase

     - Must contain at least six (6) characters.
     - The passphrase cannot include an ampersand (&) or an asterisk (*).
     - If the passphrase contains a space, it must be enclosed in quotation marks.

     Record this value. This value is required to complete registration process from the Report Server host system.

   - If you do not specify the passphrase in this step, XPSRegClient prompts you to enter and confirm it.

   - **-report**
     Specifies that a Report Server is being registered.

   - **-t timeout**
     (Optional) Specifies how long, in minutes, you have to complete the registration process from the Report Server host system. The Policy Server denies the registration request when the timeout value is reached.
     Default: 240 (4 hours)
     Limit**:** 1-1440 (one day)

- **-r retries**
  (Optional) Specifies how many failed attempts are allowed when you complete the registration process from the Report Server host system. A failed attempt can result from submitting an incorrect passphrase to the Policy Server during the registration.
  Default**:** 1
  Maximum**:** 5

- **-c comment**
  (Optional) Inserts the specified comments into the registration log file for informational purposes. Surround the comments with quotes.

- **-cp**
  (Optional) Specifies that registration log file can contain multiple lines of comments. The registration tool prompts for multiple lines of comments and inserts the specified comments into the registration log file for informational purposes. Surround the comments with quotes.

- **-l log path**
  (Optional) Specifies where the registration log file must be exported.
  **Default:** *siteminder_home*\log, where *siteminder_home* is where the Policy Server is installed.

- **-e error path**
  (Optional) Sends exceptions to the specified path.
  **Default:** stderr

- **-vT**
  (Optional) Sets the verbosity level to TRACE.

- **-vI**
  (Optional) Sets the verbosity level to INFO.

- **-vW**
  (Optional) Sets the verbosity level to WARNING.

- **-vE**
  (Optional) Sets the verbosity level to ERROR.

- **-vF**
  (Optional) Sets the verbosity level to FATAL.

The utility lists the name of the registration log file. If you did not provide a passphrase, the utility prompts for one.

4. Press Enter.
   The registration tool creates the client name and passphrase.

You can now register the Report Server with the Policy Server. Complete the registration process from the Report Server host system.

# Register the Report Server with the Policy Server

Register the Report Server with the Policy Server to create a trusted relationship between both components. Configure the connection from the Report Server host system using the Report Server registration tool.

## Gather Registration Information

Completing the registration process between the Report Server and the Policy Server requires specific information. Gather the following information before running the XPSRegClient utility from the Report Server host system.

- **Client name**
  The client name you specify using the XPSRegClient tool.
- **Passphrase**
  The passphrase you specify using the XPSRegClient tool.
- **Policy Server host**
  The IP address or name of the Policy Server host system.

## Register the Report Server

**Follow these steps:**

1. From the Report Server host system, open a command window and navigate to *report_server_home**external**scripts.*
   *report_server_home* specifies the Report Server installation location.

   **Windows**: C:\Program Files\CA\SC\CommonReporting3
   **UNIX**: /opt/CA/SharedComponents/CommonReporting3

2. Run *one* of the following commands:
   **Windows**

   ```
   regreportserver.bat -pshost host_name -client client_name -
   passphrase passphrase -psport portnum -fipsmode 0|1
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **UNIX**

   ```
   regreportserver.sh -pshost host_name -client client_name -passphrase passphrase
   -psport portnum -fipsmode 0|1
   ```

1.

- **-pshost** *host_name*
  Specifies the IP address or name of the Policy Server host system to which you are registering the Report Server.

- **-client** *client_name*
  Specifies the client name. The client name identifies the Report Server instance that you are registering.
  **Note:** This value must match the client name that you specified using the XPSRegClient utility when you registered the Report Server on the Policy Server host system.
  **Example:** If you specified "reportserver1" when using the XPSRegClient utility, enter "reportserver1".

- **-passphrase** *passphrase*
  Specifies the passphrase that is paired with the client name. The client name identifies the Report Server instance that you are registering.
  **Note:** This value must match the passphrase that you specified using the XPSRegClient utility when you registered the Report Server on the Policy Server host system.
  **Example:** If you specified CA Single Sign-On when using the XPSRegClient utility, enter CA Single Sign-On.

- **-psport** *portnum*
  (optional) Specifies the port on which the Policy Server is listening for the registration request.

- **fipsmode**
  (optional) Specifies how the communication between the Report Server and the Policy Server is encrypted.
  **Default:** 0
  0 for FIPS–compatibility mode.
  1 for FIPS–only mode.

2. Press Enter.
   You receive a message stating that the registration is successful.

# Restart the Report Server

## Restart the Report Server on Windows Systems

**Follow these steps:**

1. Click Start, Programs, BusinessObjects XI *n.n* BusinessObjects Enterprise, Central Configuration Manager.
   *n.n* is the version of BusinessObjects XI.
   The Central Configuration Manager console appears.

2. Stop the Apache Tomcat and Server Intelligence Agent Services.
   The Report Server stops.

3. Start the Apache Tomcat and Server Intelligence Agent Services.
   The Report Server is restarted.

## Restart the Report Server on UNIX Systems

**Follow these steps:**

1. Log in to the system as the non-root user that installed the Report Server.
2. Verify that at least one of the following environment variables is set to a valid utf8/UTF-8 locale:
   - LC_ALL
   - LANG
   - LC_CTYPE
3. Navigate to r*eport_server_hom*e/CommonReporting3/external/scripts and run the following command:

   ```
   . ./setupenv.sh
   ```

   *report_server_home* specifies the Report Server installation path.
4. Confirm that:
   - The IAM_RPTSRV_HOME variable is set to *report_server_home*/CommonReporting3. *report_server_home* specifies the Report Server installation path.
   - (Oracle report database only): The LD_LIBRARY_PATH variable is set to:

     ```
     $ORACLE_HOME/lib32:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
     ```

     **Example:**

     ```
     export LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$LD_LIBRARY_PATH
     ```
5. Navigate to *report_server_home*/CommonReporting3/bobje and run the following command:

   ```
   ./stopservers
   ```

   *report_server_home* specifies the Report Server installation path.
6. Shut down the Tomcat server by running the following command:

   ```
   ./tomcatshutdown.sh
   ```

   The Report Server stops.
7. Run the following command:

   ```
   ./startservers
   ```
8. Restart the Tomcat server by running the following command:

   ```
   ./tomcatstartup.sh
   ```

   The Report Server is restarted.

# Configure the Connection to the Administrative UI

Configure the Report Server and Administrative UI connection to establish a trusted relationship between both components. The Administrative UI can have a trusted relationship with one or more Policy Servers. However, each trusted relationship only allows one Report Server connection. If you have to connect to a new Report Server, delete the current Report Server connection or connect to another Policy Server to configure the connection.

Configure the connection from the Administrative UI.

**Follow these steps:**

1. Log into the Administrative UI.
2. Click Administration, Admin UI.
3. Click Report Connections, Create Report Server Connection.
   The Create Report Server Connection pane appears.
4. Type a connection name in the Connection Name field.
5. Type the Report Server host system name or IP address in the Report Server Host field.
6. Enter the Apache Tomcat connection port in the Tomcat Port field.
   This value is the web server port you entered when installing the Report Server.
7. Enter the administrator password in the respective fields.
   This value is the password you entered for the default BusinessObjects administrator account when installing the Report Server.
8. Click Submit.
   The connection between the Report Server and the Administrative UI is configured.

You have completed installing and registering the Report Server. You can now run policy analysis reports.

## Delete a Report Server Connection to the Administrative UI

Delete a Report Server connection when the connection is no longer required.

**Follow these steps:**

1. Click Administration, Admin UI.
2. Click Report Connections, Delete Report Server Connection.
3. Specify search criteria and click Search.
4. Select the connection and click Select.
   You are prompted to confirm that the connection can be deleted.

5. Click Yes.
   The Report Server connection is deleted.

# Configure an Audit Database

A Report Server connects to an audit database to create audit-based reports. Creating and managing audit reports requires a dedicated audit database.

- Register the Audit Database with the Administrative UI (see page 511)
- Audit Database and Report Server Connectivity (see page 512)

# Register the Audit Database with the Administrative UI

Register the audit database with the Administrative UI to create a trusted connection between the components. Administrators can then generate and manage audit-based reports.

The Administrative UI can have a trusted relationship with one or more Policy Servers. However, each trusted relationship only allows one audit database connection. If you have to connect to a new audit database, delete the current connection or connect to another Policy Server.

**Important!** If you are using an Oracle audit database, ensure that the user account you supply does not have the DB role. If the user account has the DB role, audit-based reports do not return correct results.

**Follow these steps:**

1. Log in to the Administrative UI.
2. Click Administration, Admin UI.
3. Click Report Connections, Create Audit Report Connection.
   The Create Audit Report Connection pane appears.
4. Select the database vendor from the Database Vendor drop-down list.
   The vendor-specific fields appear.
5. Type the name of the connection in the Connection Name field.
6. Enter the audit database host system name or IP address in the Database Server Host field.
7. Enter the audit database data source information in the DSN field:

- **Oracle**—Enter the Oracle Net Service name you specified when creating the audit database DSN.
  The service name must be associated with the DSN you entered in the Data Tab of the Policy Server Management Console when configuring the audit database connection to the Policy Server. The service name must match the DSN you create on the Report Server.

- **SQL Server**—Enter the DSN.
  The DSN name must match the DSN you specified in the Data Tab of the Policy Server Management Console when configuring the audit database connection to the Policy Server. The DSN name must also match the DSN you create on the Report Server.

8. Enter the port on which the audit database server is listening in the Database Server Port field.

9. Complete *one* of the following steps:

   - **Oracle**—Re–enter the Oracle Net Service Name in the Service Name field.

   - **SQL Server**—Enter the audit database name in the Database Name field.

10. Enter administrator credentials for the audit database in the respective fields.
    The administrator credentials must match the credentials that you specified in the Data tab of the Policy Server Management Console when configuring the audit database connection to the Policy Server.

11. Click Submit.

The audit database is registered with the Administrative UI.

# Audit Database and Report Server Connectivity

When an audit-based report is scheduled in the Administrative UI, the Administrative UI passes the following connection information to the Report Server:

- **Oracle**: The Oracle Net Service Name that identifies the audit database.

- **Microsoft SQL Server:** The name of the audit database and data source used to connect to audit database.

- The user account credentials required to access the audit database.

To configure connectivity between the audit database and the Report Server, complete the following tasks for your database:

- **Oracle**
  Confirm that the Oracle Net client is installed on the Report Server host system. Also, verify that the Oracle Net Service Name that identifies the audit database is present in the tnsnames.ora file.

- **Microsoft SQL Server:** Verify that the data source used to connect to the audit database is present on the Report Server host system.

For more information about supported database drivers, see the Platform Support Matrix.

# Generate Large Reports Successfully

Some of the Report Server services have a default timeout of 10 minutes. The Report Server can take longer than 10 minutes to generate large analysis reports. To ensure that large analysis reports are successfully generated, increase the timeout value of the Crystal Reports Job Server.

## Increase the Timeout Value on Windows

**Follow these steps:**

1. Click Start, Programs, BusinessObjects XI Release *n.n,* BusinessObjects Enterprise, Central Configuration Manager.
   The Central Configuration Manager console appears.

2. Right-click Crystal Reports Job Server and select Stop.
   The Crystal Reports Job Server service stops.

3. Right-click Crystal Reports Job Server and select Properties.
   The Crystal Reports Job Server Properties dialog appears.

4. From the Properties tab, append the following entry to the end of the string in the Command field:

   ```
   -requesttimeout 6000000
   ```

   The timeout value is measured in milliseconds. Specifying 6000000 increases the timeout value to one (1) hour.

5. Click OK.
   The Central Configuration Manager appears.

6. Right-click Crystal Reports Job Server and select Start.
   The Crystal Reports Job Server service starts.

7. Exit the Central Configuration Manager.
   The timeout value for the Crystal Reports Job Server is now increased to one hour.

## Increase the Timeout Value on UNIX

**Follow these steps:**

1. Navigate to *report_server_home*/CommonReporting3/bobje.
   *report_server_home* specifies the Report Server installation path.

2. Open the ccm.config file, and append the requesttimeout entry to the end of the *report_job_server*LAUNCH key.
   *report_job_server* is the name of the Service Intelligence Agent Node that you specified during the Report Server installation, for example, sianodeLAUNCH.

   ```
   -requesttimeout 6000000
   ```

   The timeout value is measured in milliseconds. Specifying 6000000 increases the timeout value to one hour.

   For example, the last line of the key is:

   ```
   "/opt/CA/SharedComponents/CommonReporting3/bobje/serverpids" -
   requesttimeout 6000000'
   ```

3. Save and close the file.
   The timeout value for the Crystal Reports Job Server is now increased to one hour.

# Start the Report Server

## Start the Reports Server on Windows Systems

**Follow these steps:**

1. Click Start, Programs, BusinessObjects XI *n.n*, BusinessObjects Enterprise, Central Configuration Manager.
   *n.n* is the version number of BusinessObjects XI.
   The Central Configuration Manager console appears.

2. Start the Apache Tomcat and Server Intelligence Agent Services.
   The Report Server is started.

## Start the Report Server on UNIX Systems

**Follow these steps:**

1. Log in to the system as the non-root user that installed the Report Server.

2. Verify that at least one of the following environment variables is set to a valid utf8/UTF-8 locale:

   - LC_ALL
   - LANG
   - LC_CTYPE

3. Navigate to *report_server_home*/CommonReporting3/external/scripts and run the following script:

   ```
   ../setupenv.sh
   ```

   *report_server_home* specifies the Report Server installation path.

4. Set the following variables accordingly:

   - **IAM_RPTSRV_HOME:** *report_server_home*/CommonReporting3.
   - **LD_LIBRARY_PATH** (Oracle report database only): $ORACLE_HOME/lib32:$ORACLE_HOME/lib:$CAPKIHOME/lib:$LD_LIBRARY_PATH
     **Example:** export LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME/lib:$CAPKIHOME/lib:$LD_LIBRARY_PATH
   - **CAPKIHOME:** r*eport_server_hom*e/SharedComponents/CAPKI.

   *report_server_hom*e specifies the Report Server installation path.

5. Navigate to *report_server_home*/CommonReporting3/bobje

6. Run the following command:

   ```
   ./startservers
   ```

7. Run the following command:

   ```
   ./tomcatstartup.sh
   ```

The Report Server is started.

# Stop the Report Server

## Stop the Report Server on Windows Systems

**Follow these steps:**

1. Click Start, Programs, BusinessObjects XI n.n, BusinessObjects Enterprise, Central Configuration Manager.
   *n.n* is the version of BusinessObjects XI.
   The Central Configuration Manager console appears.

2. Stop the Apache Tomcat and Server Intelligence Agent Services.
   The Report Server is stopped.

## Stop the Report Server on UNIX Systems

**Follow these steps:**

1. Log in to the system as the non-root user that installed the Report Server.
2. Verify hat at least one of the following environment variables is set to a valid utf8/UTF-8 locale:
   - LC_ALL
   - LANG
   - LC_CTYPE
3. Navigate to r*eport_server_hom*e/CommonReporting3/external/scripts and run the following command:

   ```
   ../setupenv.sh
   ```

   *report_server_home* specifies the Report Server installation path.
4. Set the following variables accordingly:
   - **IAM_RPTSRV_HOME:** *report_server_home*/CommonReporting3.
   - **LD_LIBRARY_PATH** (Oracle report database only): $ORACLE_HOME/lib32:$ORACLE_HOME /lib:$ETPKIHOME/lib:$LD_LIBRARY_PATH
     **Example:** export LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$ORACLE_HOME /lib:$ETPKIHOME/lib:$LD_LIBRARY_PATH
   - **ETPKIHOME**: r*eport_server_hom*e/CommonReporting3.

   *report_server_hom*e specifies the Report Server installation path.
5. Navigate to r*eport_server_home*/CommonReporting3/bobje.
   *report_server_home* specifies the Report Server installation path.
6. Run the following command:

   ```
   ./stopservers
   ```
7. Run the following command:

   ```
   ./tomcatshutdown.sh
   ```

The Report Server is stopped.

# Uninstall the Report Server

Uninstall the Report Server and its components to remove it from your system.

- Uninstall the Report Server Configuration Wizard from Windows (see page 517)
- Uninstall the Report Server Configuration Wizard from UNIX (see page 517)
- Uninstall the Report Server from Windows (see page 518)
- Uninstall the Report Server from UNIX (see page 519)

# Uninstall the Report Server Configuration Wizard from Windows

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to *report_server_home* \CommonReporting3\Uninstall_CA_SiteMinder_ConfigurationWizard.
   *report_server_home* specifies the Report Server installation path.

   **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.
3. Double-click uninstall.exe.
4. Follow the prompts.
   If a message prompts you to remove shared files, click No to All.
5. If requested, restart the system.
   The Report Server Configuration Wizard is uninstalled.

# Uninstall the Report Server Configuration Wizard from UNIX

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to *report_server_home/*CommonReporting3 /Uninstall_CA_SiteMinder_ConfigurationWizard.
   *report_server_home* specifies the Report Server installation path.
3. Run the following command:

   ```
   ./uninstall
   ```
4. Follow the prompts.
   The Report Server Configuration Wizard is uninstalled.

# Uninstall the Report Server from Windows

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to r*eport_server_home*\CommonReporting3\Uninstall CA Business Intelligence 3.3.
   *report_server_home* specifies the Report Server installation path.
3. Double-click Uninstall CA Business Intelligence 3.3.exe

   **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

   The uninstaller starts.
4. Follow the prompts.
5. If requested, reboot the system.
   The Report Server is uninstalled.

**Note:** Uninstalling the Report Server does not remove the tables in the report database. Manually remove these tables.

**Uninstall the Report Server using a Wizard**

If you installed the Report Server using an unattended installation, the uninstaller starts silently. To uninstall the server using a wizard, complete these steps:

1. Open a command prompt.

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.
2. Navigate to r*eport_server_home*\CommonReporting3\Uninstall CA Business Intelligence 3.3.
   *report_server_home* is the specifies the Report Server installation path.
3. Enter the following command:

   ```
   Uninstall CA Business Intelligence 3.3.exe -i swing
   ```

4. Follow the prompts.
5. If requested, reboot the system.

   The Report Server is uninstalled.

# Uninstall the Report Server from UNIX

Do not manually remove the installation directories to uninstall this component. Execute the uninstall shell script. If you only remove the installation directories, related registries can remain. If you try to reinstall this component on this host system, the entries can prevent a successful installation.

**Follow these steps:**

1. Open a Bourne shell and navigate to *report_server_home*/CommonReporting3/Uninstall.
   *report_server_home* specifies the Report Server installation path.
2. Run the following command:

   ```
   ./UninstallCABusinessIntelligence 3.3
   ```

3. Follow the prompts.
   The Report Server is uninstalled.

**Note:** Uninstalling the Report Server does not remove the tables in the report database. Manually remove these tables.

# Remove Remaining Components from a Windows System

Manually remove leftover items to keep the system as clean as possible. If you reinstall the Report Server to the same system, removing leftover items prevents the Report Server installation from failing.

**Follow these steps:**

1. Navigate to r*eport_server_home*.
   *report_server_home* specifies the Report Server installation path.
2. Delete the following directory:
   CommonReporting3
   You have removed the leftover items.

# Remove Remaining Components from a UNIX System

Manually remove leftover items to keep the system as clean as possible. If you reinstall the Report Server to the same system, removing leftover items prevents the Report Server installation from failing.

**Important!** Other CA products can share the SharedComponents directory. The profile.CA file sets the environment variables for this location. Be sure that no other CA products are sharing the SharedComponets directory before you remove this file.

**Follow these steps:**

1. Navigate to r*eport_server_home*
   *report_server_home* specifies the Report Server installation path.

2. Delete the following folder:
   CommonReporting

3. Navigate to /etc.

4. Remove the **profile.CA** file.
   You have removed the leftover items.

# Remove the Report Database Tables

Uninstalling the Report Server does not remove the tables in the report database. Access the report database and manually remove all tables.

# Install Agents

This section explains how to install Agents:



# Policy Server Preparation for the Web Agent Installation

Before you install a Web Agent, you must have:

- Installed the Policy Server.
- Configured a policy/key store to communicate with the Policy Server.
- Installed and registered the Administrative UI.
- Confirmed that the Policy Server can communicate with the system on which you will install the Web Agent.

Before you can register a trusted host at the Web Agent site, the following objects must be configured in the Administrative UI.

To centrally manage Agents, configure the following using the Administrative UI:

- **A CA Single Sign-On Administrator that has the right to register trusted hosts**—A trusted host is a client computer where one or more CA Single Sign-On Web Agents are installed. The term trusted host refers to the physical system. There must be an administrator with the permission to register trusted hosts. The default CA Single Sign-On administrator has this permission.
- **Agent identity—**An Agent identity establishes a mapping between the name and the IP address of the web server instance hosting a Web Agent. You define an Agent identity from the Agents object in the Administrative UI. You assign the Agent identity a name and specify the Agent type as a Web Agent.

**Note:** The name you assign for the Agent is the same name you specify in the DefaultAgentName parameter for the Agent Configuration Object.

- **Host Configuration Object—**A host configuration object defines the communication between the trusted host and the Policy Server after the initial connection between the two is made.
  Do not confuse the host configuration object with the trusted host configuration file, SmHost.conf, which is installed at the trusted host after a successful host registration. The settings in the SmHost.conf file let the host connect to a Policy Server for the first connection only. Subsequent connections are governed by the host configuration object.

- **Agent Configuration Object—**An Agent configuration object includes the parameters that define the Web Agent configuration. There are a few required parameters you are required to set for the basic operation described below.

  **Note:** If you plan to use the NTLM authentication scheme, or enable the Windows User Security Context feature, do not specify values for these IIS Web Agent parameters.

  - **For all Agents—**The Agent Configuration Object must include a value for the DefaultAgentName. The DefaultAgentName must match the Agent identity name you specified in the Agents object. The DefaultAgentName identifies the Agent identity that the Web Agent uses when it detects an IP address on its web server that does not have an Agent identity assigned to it.

  - **For Domino Web Agents—**The Agent Configuration Object must include values for the following parameters:

    - **DominoDefaultUser—**If the user is not in the Domino Directory, and they have been authenticated by CA Single Sign-On against another user directory, this is the name by which the Domino web agent identifies that user to the Domino server. The DominoDefaultUser value can be encrypted.

    - **DominoSuperUser—**Ensures that all users successfully logged into CA Single Sign-On are logged into Domino as the DominoSuperUser. The DominoSuperUser value can be encrypted.

  - **For IIS Web Agents—**The Agent Configuration Object must include values for the DefaultUserName and DefaultPassword parameters. The DefaultUserName andDefaultPassword identify an existing Windows account that has sufficient privileges to access resources on an IIS web server protected by CA Single Sign-On. When users need to access resources on an IIS web server protected by CA Single Sign-On, they may not have the necessary server access privileges. The Web Agent must use the Windows account, which is previously assigned by an administrator, to act as a proxy user account for users granted access by CA Single Sign-On.

# Web Agent for Apache

The following sections detail how to install and configure an agent on Apache-based web servers.

## Policy Server Requirements for Apache-based Servers

Verify the following criteria:

- Your Policy Server is installed and configured.
- Your Policy server can communicate with the computer where you plan to install the agent.

To install and configure an agent, a Policy Server requires at least the following items:

- An administrator that has the right to register trusted hosts.
  A trusted host is a client computer where one or more agents are installed and registered with the Policy Server. The administrator must have permissions to register trusted hosts with the Policy Server. Registering a trusted host creates a unique trusted host name object on the Policy Server.
- An Agent identity
  An Agent identity establishes a mapping between the Policy Server and the name or IP address of the web server instance hosting an Agent. Define an Agent identity from the Agents object in the Administrative UI. You assign it a name and specify the Agent type as a Web Agent.
- A Host Configuration Object (HCO)
  The host configuration object on the Policy Server defines the communication between the agent and the Policy Server that occurs after an initial connection. The Initial connections use the parameters in the SmHost.conf file.
- Agent Configuration Object (ACO)
  This object includes the parameters that define the agent configuration. All agents require at least one of the following configuration parameters that are defined in the ACO:
  - **AgentName**
    Defines the identity of the web agent. This identity links the name and the IP address or FQDN of each web server instance hosting an Agent.
    The value of the DefaultAgentName is used instead of the AgentName parameter if any of the following events occur:
    - The AgentName parameter is disabled.
    - The value of AgentName parameter is empty.
    - The values of the AgentName parameter do *not* match any existing agent object.

> **Note:** This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add each value to a separate line in the file.

**Default:** No default
**Limit**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
**Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.
**Example:** myagent1,192.168.0.0 (IPV4)
**Example:** myagent2, 2001:DB8::/32 (IPV6)
**Example:** myagent,www.example.com
**Example** (multiple AgentName parameters): AgentName1, AgentName2, AgentName3. The value of each AgentName*number* parameter is limited to 4,000 characters.

- **DefaultAgentName**
  Defines a name that the agent uses to process requests. The value for DefaultAgentName is used for requests on an IP address or interface when no agent name value exists in the AgentName parameter.
  If you are using virtual servers, you can set up your CA Single Sign-On environment quickly by using a DefaultAgentName. Using DefaultAgentName means that you do not need to define a separate agent for each virtual server.

  > **Important!** If you do not specify a value for the DefaultAgentName parameter, then the value of the AgentName parameter requires every agent identity in its list. Otherwise, the Policy Server cannot tie policies to the agent.

  **Default:** No default.
  **Limit**: Multiple values are allowed.
  **Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.

# Hardware Requirements for an Apache-based Agent

- **Windows operating environment requirements**
  CA Single Sign-On agents operating on Windows operating environments require the following hardware:
  - CPU: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:

- 2-GB free disk space in the installation location.
- .5-GB free disk space in the temporary location

- **UNIX operating environment requirements**
  CA Single Sign-On agents operating on UNIX operating environments require the following hardware:
  - CPU:
    - Solaris operating environment: SPARC
    - Red Hat operating environment: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in /temp.

**Note**: Daily operation of the agent requires 10 MB of free disk space in /tmp. The agent creates files and named pipes under /tmp. The path to which these files and pipes are created cannot be changed.

# Apache-based Server Preparations for Windows

**Contents**



## Install an Apache Web Server on Windows as a Service for All Users

When an Apache-based web server is installed using a single user account, the Agent configuration cannot detect the Apache-based web server installation.

To correct this problem, select the following option when you install an Apache-based web server on a Windows operating environment:

```
"install as a service, available for all users".
```

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based Web Agents

For Agents for Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

# Apache-based Server Preparations on UNIX

**Contents**



## Set the DISPLAY For CA Single Sign-On Agent Installations on UNIX

If you are installing the CA Single Sign-On Agent on a UNIX system from a remote terminal, such as a Telnet or Exceed terminal, be sure the DISPLAY variable is set for the local system. For example, if your machine is 111.11.1.12, set the variable as follows:

DISPLAY=111.11.1.12:0.0

export DISPLAY

**Note:** You can also install the agent using the console mode installation, which does not require the X window display mode.

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based CA Single Sign-On Agents

For any agents for Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

## Required Solaris Patches

Before installing a CA Single Sign-On Agent on a Solaris computer, install the following patches:

- **Solaris 9**
  Requires patch 111711-16.
- **Solaris 10**
  Requires patch 119963-08.

You can verify installed patch versions by logging in as the root user and executing the following command:

```
showrev -p | grep patch_id
```

To locate Solaris patches, go to the Oracle Solution Center.

## AIX Requirements

CA Single Sign-On agents running on AIX systems require the following components:

- To run a rearchitected (framework) CA Single Sign-On Apache-based agent on an AIX system, your C/C++ runtime environment must be version 8.0.0.0.

# Apache-based Server Preparations for Linux

This content describes the preparations that are required to prepare for an Apache Web Agent install on a Red Hat Enterprise Linux system.



## Required Software Packages

The following software packages are required to install Web Agents on 64-bit Linux systems:

- Binutils 2.17
- GCC 4.7.2

## Required Linux Libraries

CA Single Sign-On requires certain Linux libraries for components that operate on Linux. We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

The following list describes the commands for installing the required libraries on the host system:

**Red Hat 5.x**

```
yum install -y compat-gcc-34-c++
yum install -y libidn.so.11
yum install -y libstdc++.i686
yum install -y  ncurses-libs.i68
```

**Red Hat 6.x**

```
yum install -y libstdc++.i686
yum install -y libidn.so.11
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y ncurses-libs.i686
yum install -y libXrender.i686
yum install -y libXtst.i686
```

**Additional Packages for Red Hat 6.x 64-bit**

```
yum install -y libXau.i686
yum install -y libXext.i686
yum install -y libxcb.i686
yum install -y compat-libstdc++-33.i686
yum install -y compat-db42.i686
yum install -y compat-db.i686
yum install -y compat-db43.i686
yum install -y libXi.i686
yum install -y libX11.i686
yum install -y libXtst.i686
yum install -y libXrender.i686
yum install -y libXft.i686
yum install -y libexpat.so.1
yum install -y libXt.i686
yum install -y libfreetype.so.6
yum install -y libXp.i686
yum install -y libfontconfig.so.1
yum install -y libstdc++.i686
yum install -y libICE.i686
yum install -y compat-libtermcap.i686
yum install -y libidn.i686
yum install -y libSM.i686
yum install -y libuuid.i686
```

**Red Hat 7.x**

```
yum install -y libstdc++.i686
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y libXrender.i686
yum install -y libidn.so.11
yum install -y libXtst.i686
yum install -y ncurses-libs.i686
```

If the correct library is unavailable, CA Single Sign-On displays the following error:

```
java.lang.UnsatisfiedLinkError
```

## Install Red Hat Legacy Software Development Tools

Install all the items included in the Red Hat Legacy Software Development tools package. You require these tools to compile the Apache Web Server.

## Compile an Apache Web Server on a Linux System

For the CA Single Sign-On Agent to operate with an Apache web server running Linux, compile the server. Compiling is required because the Agent code uses pthreads (a library of POSIX-compliant thread routines), but the Apache server on the Linux platform does not, by default.

If you do not compile with the lpthreadoption, the Apache server starts up, but then hangs and does not handle any requests. The Apache server on Linux cannot initialize a module which uses pthreads due to issues with the Linux dynamic loader.

**Follow these steps:**

1. Enter the following commands:

```
LIBS=-lpthread
export LIBS
```

2. Configure Apache as usual by entering the following commands:

```
configure --enable-module=so --prefix=your_install_target_directory
make
make install
```

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based Agents

For agents running on Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

# IBM HTTP Server Preparations

## Enable Write Permissions for IBM HTTP Server Logs

If you install the CA Single Sign-On Agent on an IBM HTTP Server, this web server gets installed as root and its subdirectories do not give all users in all groups Write permissions.

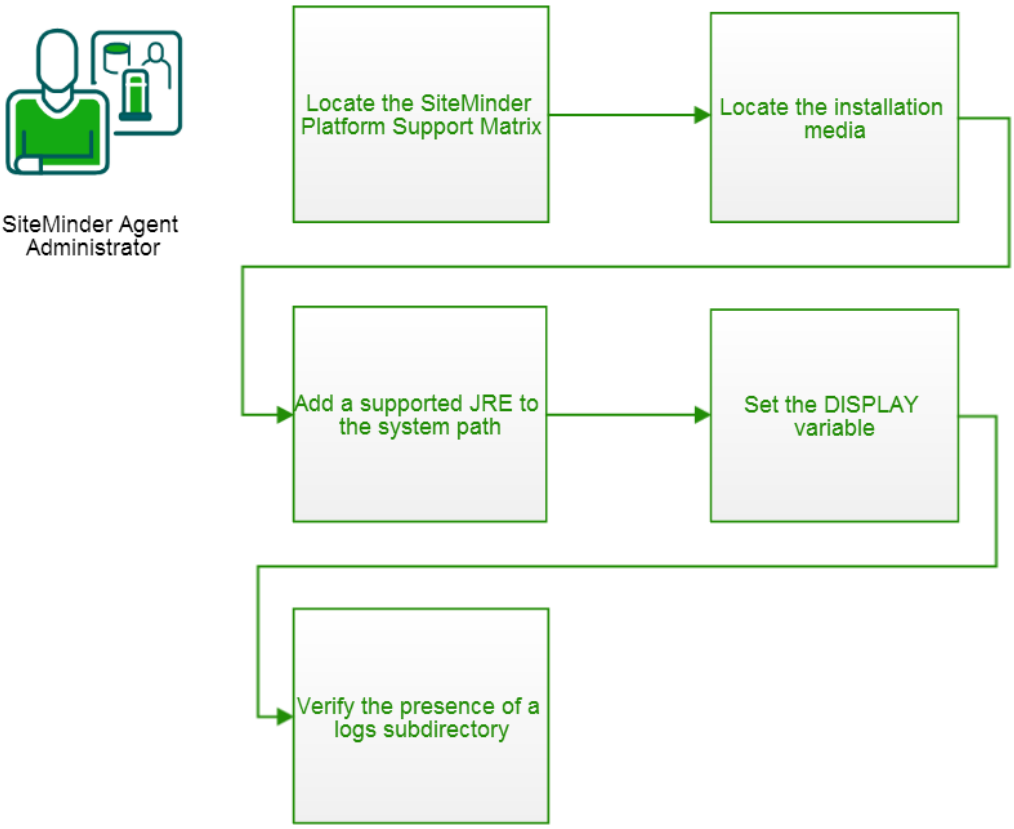
For the Low Level Agent Worker Process (LLAWP) to write agent initialization messages to the web server logs, the user running the web server needs permission to write to the web server's log directory. Ensure that you allow write permissions for this user.

# Preparations for z/OS

**Contents**



Before you install and configure a CA Single Sign-On agent on the z/OS operating environment, perform the preparation steps described in this process.




## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Set the DISPLAY Variable for CA Single Sign-On Agent Installations on z/OS

If you are installing the CA Single Sign-On agent on a z/OS system from a remote terminal, verify that the DISPLAY variable is set for the local system. For example, if your server IP address is 111.11.1.12, set the variable as follows:

```
export DISPLAY=111.11.1.12:0.0
```

**Note:** You can also install the CA Single Sign-On agent using the console mode installation, which does not require the X window display mode.

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based CA Single Sign-On Agents

For any agents for Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

## Add a Supported JRE to the System Path

On z/OS systems, before installing the CA Single Sign-On agent, verify that a supported JRE is present on the system and defined in the PATH and JAVA_HOME system variables.

**Follow these steps:**

Enter the following commands at a command prompt:

```
export PATH=JRE/bin:$PATH
export JAVA_HOME=JRE
```

- ***JRE***
  Specifies the location of the JRE.
  For example, /sys/java64bt/v6r0m1/usr/lpp/java/J*version_number*

# Install and Configure Apache-based Agents on Windows

This section contains the following topics:



## How to Install Apache-based Agents on Windows

**Contents**



### Gather the Information for the Installation Program

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path

### Run the Installation Program on Windows

The installation program for the agent installs the agent on one computer at a time using the Windows operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation programs also create a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy the Web Agent installation executable file to a temporary directory on your web server.
2. Do *one* of the following steps:
   - For wizard-based installations, right-click the installation executable file, and then select Run as Administrator.
   - For console-based installations, open a command line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```
3. Use the information that you gathered previously to complete the installation.

## How to Configure Apache-based Agents on Windows

**Contents**



Configuring the agent occurs after the installation. Configuration requires several separate procedures which are described using the following process:

### Gather the Information Required by the Configuration Program on Windows

Gather the following information about the environment for the product before running the configuration program for the agent:

- **Register Host**
  Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.
  **Default**:Yes
  **Options**: Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This CA Single Sign-On user account requires privileges to register trusted hosts.

- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.

- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.

- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.
    **Default**: FIPS Compatibility/AES Compatibility
    FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

**Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the CA Single Sign-On agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  - **Default: SmHost.conf**
- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config
- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.
- **Select Servers**
  This step has multiple screens. The first screen indicates the server type (Apache), and the next screen displays the web server instances that the configuration program finds on the computer. Select the check boxes of the server type, and the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.
- **Apache 2.4.x Install Location**
  Specifies the location of the installation directory for your Apache-based server (version 2.4 or higher).
- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj
- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Run the Web Agent Configuration Program on Windows

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:

   ```
   web_agent_home\install_config_info
   ```

- ***web_agent_home***
  Indicates the directory where the CA Single Sign-On agent is installed on your web server.
  **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
  **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
  **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32

2. Use *one* of the following configuration methods:
   - For a GUI-based configuration, go to Step 3.
   - For a console-based configuration, go to Step 5.
3. Right-click the following executable, and then select Run as Administrator:

   ```
   ca-wa-config.exe
   ```

4. Go to Step 8.
5. Open a Command Prompt window with Administrator privileges.
6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```

7. Go to Step 8.
8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## Run a Silent Installation and Configuration for Apache Agents on Windows

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The CA Single Sign-On Web Agent Installation wizard.
   b. The CA Single Sign-On Web Agent Configuration wizard.
2. Locate the following file on your first web server:

```
web_agent_home\install_config_info\ca-wa-installer.properties
```

**Note**: If the path contains spaces, surround it with quotes.

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On agent is installed on your web server.
   **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
   **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
   **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32

4. Perform each of the following steps on the other web servers in your environment:

**Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.

   b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:

      - The CA Single Sign-On Web Agent Installation executable file.
      - CA Single Sign-On ca-wa-installer properties file.

   c. Open a Command Prompt window with Administrative privileges in the temporary directory.

   d. Run the following command:

      ```
      agent_executable -f properties_file -i silent
      ```

      The CA Single Sign-On agent is installed and configured on the subsequent server silently.

   e. (Optional) Delete the temporary directory from your subsequent web server.

5. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Install and Configure Apache-based Agents on UNIX/Linux

This section contains the following topics:

# How to Install Apache-based Agents on UNIX or Linux

**Contents**



## Gather the Information for the Installation

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path.

## Run the Installation Program on UNIX/Linux

The installation program for the CA Single Sign-On agent installs the agent on one computer at a time using the UNIX or Linux operating environments. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy CA Single Sign-On agent installation executable file to a temporary directory on your web server.
2. Log in as a root user.
3. Do *one* of the following steps:
   - For wizard-based installations, run the installation executable file.
   - For console-based installations, open a command-line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```

4. Use the information from your agent Installation worksheet to complete the installation program.

## How to Configure Apache-based Agents on UNIX or Linux

**Contents**



### Gather the Information that the Configuration Program Requires on UNIX/Linux

Gather the following information about the environment for the product before running the configuration program for the agent:

- **Register Host**

Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.

**Default**:Yes

**Options**: Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This CA Single Sign-On user account requires privileges to register trusted hosts.

- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.

- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.

- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.

  - **Default: FIPS Compatibility/AES Compatibility**

  FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

  **Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the CA Single Sign-On agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Edit the configuration files for embedded Apache web servers on RedHat Linux

For an embedded Apache web server (included by default) on a RedHat Linux system, modify certain configuration files to accommodate the product first.

**Follow these steps:**

1. Log on to the RedHat Linux system.
2. Open the following file with a text editor:

   *web_agent_home*/ca_wa_env.sh

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On Agent is installed.
   **Default** (UNIX/Linux installations): /opt/ca/webagent
4. Verify that the line that sets the LD_PRELOAD variable begins with a comment (the LD_PRELOAD variable setting is disabled).
5. Save the changes and close the ca_wa_env.sh file.
6. Open the following file with a text editor:

   /etc/sysconfig/httpd

7. Add the following line to the end of the file:

   PATH=$PATH:web_agent_home/bin

8. Save the changes and close the text editor.

## Source the Agent Environment Script on UNIX or Linux

The agent installation program creates an environment script, **ca_wa_env.sh** in the following directory:

*web_agent_home*/ca_wa_env.sh

*web_agent_home* indicates the directory where the Agent is installed. The default UNIX/LINUX location for the script is:

```
opt/ca/webagent
```

For RHEL 7, include the content of the source script in the directory:

```
/etc/sysconfig/httpd
```

The following is a sample of the modified script in the directory /etc/sysconfig/httpd. Strings in **bold** are in effect and others are commented out.

Note the following:

- Replace any ${*VARIABLE*} with the actual value.
- To determine the values for the variables ${LD_LIBRARY_PATH} and ${PATH}, use the **env** command before you add the script contents.

```
NETE_WA_ROOT=/opt/CA/webagent
#export NETE_WA_ROOT
NETE_WA_PATH=/opt/CA/webagent/bin
#NETE_WA_PATH=${NETE_WA_ROOT}/bin
#export NETE_WA_PATH
 CAPKIHOME=/opt/CA/webagent/CAPKI#export CAPKIHOME
LD_LIBRARY_PATH=/opt/CA/webagent/bin:/opt/CA/webagent/bin/thirdparty (http://bin/opt/CA
/webagent/bin/thirdparty)#LD_LIBRARY_PATH=${NETE_WA_ROOT}/bin:${NETE_WA_ROOT}/bin
/thirdparty:${LD_LIBRARY_PATH}
#export LD_LIBRARY_PATH
PATH=/opt/CA/webagent/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr
/bin:/root/bin (http://bin/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root
/bin)
#PATH=/opt/CA/webagent/bin:${PATH}
#PATH=${NETE_WA_PATH}:${PATH}
#export PATH
```

For most Apache-based web servers, source this script *before* doing any of the following tasks:

- Running the agent configuration program.
- Starting the web server.

> **Note**: If you perform *all* the previous tasks in the *same* shell, only source the script *once*.

For the embedded Apache web server included with RedHat Linux, do *one* of the following tasks:

- Source the script *before* starting the httpd service.
- Source the script in the following file instead of starting it manually each time:

  ```
  /etc/init.d/htppd
  ```

## Set the Library Path Variable on UNIX or Linux

Set the library path variable on UNIX or Linux systems before running the agent configuration program.

The following table lists the library path variables for the various UNIX and Linux operating environments:

| Operating System | Name of Library Path Variable |
|---|---|
| AIX | LIBPATH |
| Linux | LD_LIBRARY_PATH |
| Solaris | LD_LIBRARY_PATH |

Set the value of the library path variable to the *agent_home*/bin directory.

- ***agent_home***
  Indicates the directory where the Agent is installed.

## Set Web Agent Variables when using apachectl Script

You run your Apache server using the apachectl script (such as when running an Apache web server on POSIX). Adding a line to the apachectl script sets the environment variables for the agent.

**Follow these steps:**

1. Locate a line resembling the following example:

   ```
   # Source /etc/sysconfig/httpd for $HTTPD setting, etc
   ```

2. Add the following line *after* the line in the previous example:

   ```
   . web_agent_home/ca_wa_env.sh
   ```

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On Agent is installed.
   **Default** (UNIX/Linux installations): /opt/ca/webagent

## Run the Web Agent Configuration Program on UNIX/Linux

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:

   *web_agent_home*/install_config_info

   web_agent_home Indicates the directory where the CA Single Sign-On Agent is installed.
   Default (UNIX/Linux installations): /opt/ca/webagent

2. Use *one* of the following configuration methods:

- For a GUI-based configuration, go to Step 3.
- For a console-based configuration, go to Step 5.

3. Run the following executable file:

   ```
   ca-wa-config.bin
   ```

4. Go to Step 8.
5. Open a Command Prompt window with root privileges.
6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```

7. Go to Step 8.
8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.

   The agent runtime instance is created for your web servers.

## Set the LD_ASSUME_KERNEL for Apache Agent on SuSE Linux 9 for zSeries

After you install the Web Agent on an Apache web server running on SuSE Linux 9 for zSeries, set the LD_ASSUME_KERNEL environment variable as follows:

LD_ASSUME_KERNEL=2.4.21

export LD_ASSUME_KERNEL

**Important!** You must set this variable to 2.4.21 because it represents the kernel release upon which the Web Agent libraries are built.

Without this setting, the following problems occur:

- The Apache web server will not start properly.
- Host registration dumps core.

## Set the CAPKIHOME Variable for Red Hat Linux 6 Systems

If you want to run an Apache-based Web Agent on a Red Hat Linux system, set the CAPKIHOME environment variable by entering the following commands:

```
CAPKIHOME="/usr/local/CA/webagent/CAPKI"
export CAPKIHOME
```

# Optional Agent Settings for UNIX/Linux

**Contents**



## Set Web Agent Variables when using apachectl Script

You run your Apache server using the apachectl script (such as when running an Apache web server on POSIX). Adding a line to the apachectl script sets the environment variables for the agent.

**Follow these steps:**

1. Locate a line resembling the following example:

   ```
   # Source /etc/sysconfig/httpd for $HTTPD setting, etc
   ```

2. Add the following line *after* the line in the previous example:

   ```
   . web_agent_home/ca_wa_env.sh
   ```

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On Agent is installed.
   **Default** (UNIX/Linux installations): /opt/ca/webagent

## Improve Server Performance with Optional httpd.conf File Changes

You can improve server performance by modifying the default configuration settings in the httpd.conf file; however, these changes are *not* required:

**Follow these steps:**

1. For Apache- based servers, assign a higher priority level to your Apache20WebAgent.dll file than any other auth or access modules on your web server.

2. For low-traffic websites, define the following directives:

   - Set MaxRequestsPerChild>1000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >5
   - MaxSpareServers>10
   - StartServers=MinSpareServers>5

3. For high-traffic websites, define the following directives:

   - Set MaxRequestsPerChild>3000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >10
   - MaxSpareServers>15

- StartServers=MinSpareServers>10

**Note:** CA Services can provide assistance with performance-tuning for your particular environment.

## Run a Silent Installation and Configuration for Apache-based Agents on UNIX or Linux

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The CA Single Sign-On Web Agent Installation wizard.
   b. The CA Single Sign-On Web Agent Configuration wizard.
2. Locate the following file on your first web server:

   ```
   web_agent_home/install_config_info/ca-wa-installer.properties
   ```

   **Note**: If the path contains spaces, surround it with quotes.

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On agent is installed on your web server.
4. Perform each of the following steps on the other web servers in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.
   b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:

- The CA Single Sign-On Web Agent Installation executable file.
- CA Single Sign-On ca-wa-installer properties file.

c. Open a Command Prompt window with Administrative privileges in the temporary directory.

d. Run the following command:

```
agent_executable -f properties_file -i silent
```

The CA Single Sign-On agent is installed and configured on the subsequent server silently.

e. (Optional) Delete the temporary directory from your subsequent web server.

5. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Install and Configure Apache-based Agents on z/OS

This section contains the following topics:

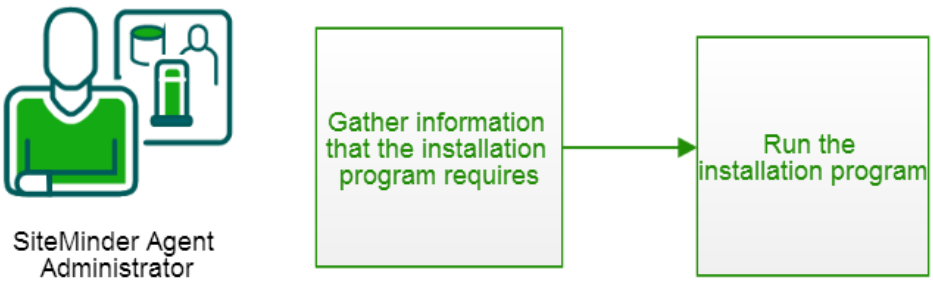
- How to Install Agents on z/OS Systems (see page 547)
- How to Configure Agents on z/OS Systems (see page 549)

## How to Install Agents on z/OS Systems

**Contents**



To Install CA Single Sign-On agents on the z/OS operating environments, perform the following process.



## Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Run the CA Single Sign-On Agent Installation Program on z/OS

The installation program for the CA Single Sign-On agent installs the agent on a single computer running the z/OS operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

You install the CA Single Sign-On agent using the installation media on the Technical Support site.

**Note:** Verify that you have executable permissions. To add executable permissions to the installation media, run the following command:

```
chmod +x installation_media
```

- ***installation_media***
  Specifies the CA Single Sign-On agent installer executable.

**Follow these steps:**

1. Log in as a root user.
2. Exit all applications that are running.
3. Open a shell and navigate to the installation media.
4. Run the installation program in GUI or console mode by entering one of the following commands:
   GUI Mode:

   ```
   java -jar  installation_media
   ```

   Console Mode:

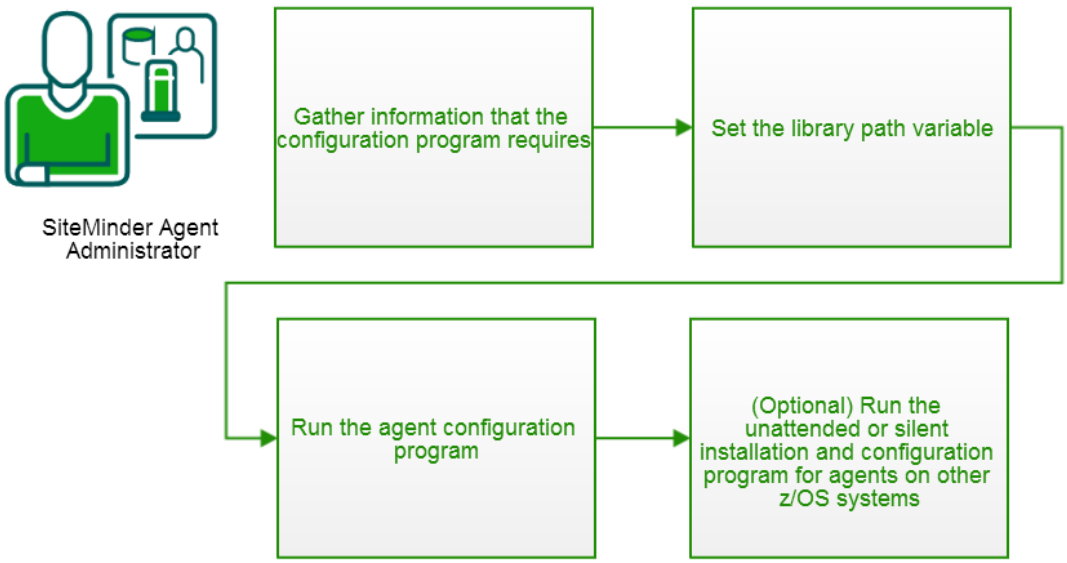
   ```
   java -jar installation_media -i console
   ```
5. Use the information from that you gathered earlier to complete the installation program.

# How to Configure Agents on z/OS Systems

**Contents**



Configure the CA Single Sign-On agent after installation by performing the procedures described in the following process.




## Gather the Information that the Configuration Program Requires on z/OS

Gather the following information about the environment for the product before running the configuration program for the agent:

Register Host

Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to CA Single Sign-On Policy Servers when it starts. Register each agent instance as a trusted host only once.

**Default:**Yes

**Options:** Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in thePolicy Server. This CA Single Sign-On user account requires privileges to register trusted hosts.

- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.

- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.

- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.

  - **Default: FIPS Compatibility/AES Compatibility**

**Note:** FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

**Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the CA Single Sign-On agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the CA Single Sign-On Policy Server uses to encrypt communications to the Web Agents.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **IBM HTTP Server for z/OS Path**
  Specifies the path of the IBM HTTP Server httpd.conf configuration file.
  **Default**: The home directory of the account that is used to run the configuration wizard.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Set the Library Path Variable on z/OS

Set the library path variable on z/OS systems before running the agent configuration program.

```
export LIBPATH=web_agent_home/bin
```

- ***web_agent_home***
  Indicates the directory where the CA Single Sign-On Agent is installed.
  **Default**: /opt/ca/webagent

## Run the CA Single Sign-On Agent Configuration Program on z/OS

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other z/OS systems in the future.

**Note:** Verify that you have executable permissions. To add executable permissions to the installation media, run the following command:

```
chmod +x installation_media
```

- ***installation_media***
  Specifies the CA Single Sign-On agent installer executable.

**Follow these steps:**

1. Log in as a root user.
2. Exit all applications that are running.
3. Open a shell and navigate to the following directory:

   ```
   web_agent_home/install_config_info
   ```

4. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On Agent is installed.
   **Default** (UNIX/Linux installations): /opt/ca/webagent
5. Run the configuration program in GUI or console mode by entering one of the following commands:
   GUI Mode:

   ```
   ca-wa-config.sh
   ```

   Console Mode:

   ```
   ca-wa-config.sh -i console
   ```

6. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## (Optional) Run the Unattended or Silent Installation and Configuration Programs for CA Single Sign-On Agents on z/OS

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
    a. The CA Single Sign-On agent Installation wizard.
    b. The CA Single Sign-On agent configuration wizard.
2. Locate the following file on your first web server:

    ```
    web_agent_home/install_config_info/ca-wa-installer.properties
    ```

    **Note**: If the path contains spaces, surround it with quotes.

3. ***web_agent_home***
    Indicates the directory where the CA Single Sign-On Agent is installed.
    **Default** (UNIX/Linux installations): /opt/ca/webagent
4. Perform each of the following steps on the subsequent web server:

    **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

    a. Create a temporary directory on the subsequent web server.
    b. Copy the following files from the web server where you ran the wizards (from Steps 1 and 2) to the temporary directory on your subsequent web server:
        - The CA Single Sign-On Agent Installation executable file.
        - The ca-wa-installer properties file.
    c. Open a Command Prompt window with root privileges in the temporary directory.
    d. Run the following command:

        ```
        java -jar installation_media -f ca-installer.properties -i silent
        ```

        - ***installation_media***
            Specifies the CA Single Sign-On Agent installer executable.

The CA Single Sign-On agent is installed and configured on the web server silently.

e. (Optional) Delete the temporary directory from your web server.

5. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Uninstall an Agent from an Apache-based Server on Windows

Before you un–install the CA Single Sign-On Web Agent from a Windows operating environment, consider making backup copies of your registry settings and Web Agent configuration settings.

Consider the following items:

- All Web Agents for all installed web servers are uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) are removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories could possiblycontain customized files.

**Follow these steps:**

1. Stop the web server.
2. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
3. Choose *one* of the following procedures:
   - To remove the Web Agent using the wizard, go to Step 4.
   - To remove the Web Agent using the console-based program, go to Step 9.
4. Click Start, Control Panel, Programs and Features.
   A list of installed programs appears.
5. Click CA Single Sign-On Web Agent *version_number*.
6. Click Uninstall/Change.
   The uninstallation wizard appears.
7. Review the information in the Uninstall CA Single Sign-On Web Agent dialog, then click Uninstall.
   The wizard removes the web agent.
8. Wait for the wizard to finish, then go to Step 13.
9. Open a command-line window.
10. Navigate to the following directory.

```
web_agent_home
```

- ***web_agent_home***
  Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
  **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
  **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
  **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

11. Run the following command:

    ```
    ca-wa-uninstall.cmd -i console
    ```

12. Wait for the un–installation program to finish, then go to Step 13.

13. Start the web server.

**Important!** Delete the ZeroG registry file from the following location after uninstalling the Web Agent: C:\Program Files\ZeroG Registry\com.zerog.registry.xml

# Uninstall an Apache-based Agent from a UNIX System

These instructions are for GUI and Console Mode removal.

**Note:** Removing a Web Agent from a 64-bit SuSE Linux 10 system requires additional preparations.

Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

The steps for the two modes are the same, with these exceptions for Console Mode:

- Select the option that you want by entering a corresponding number.
- Press Enter after each step to proceed through the process instead of "clicking Next," as stated in the following procedure.

**Note:** Before you uninstall, we recommend copying your agent configuration settings to have as a backup.

**Follow these steps:**

1. Stop the web server.
2. Log in to the UNIX system.
3. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
4. Navigate to the directory where the Web Agent is installed:

   `web_agent_home/install_config_info/ca-wa-uninstall`
5. If necessary, verify that you have execute permissions on the uninstallation program by entering chmod +x uninstall.
6. From a console window, enter one of the following commands:
   - GUI mode: ./uninstall
   - Console mode: ./uninstall -i console

   The uninstallation program starts.
7. Read the information in the dialog to confirm the removal of the Web Agent, then click Uninstall. The Web Agent is removed from the system.
8. Click Done to exit the uninstallation program.
9. (Optional) For Apache-based agents, remove the lines from the httpd.conf file that the Configuration Wizard added.
10. Change to your home directory (the current directory has been deleted).
11. Restart the web servers.

# Uninstall an Apache-based Agent from a z/OS System

These instructions are for GUI and Console Mode removal. The steps for the two modes are the same, with these exceptions for Console Mode:

- Select the option that you want by entering a corresponding number.
- Press Enter after each step to proceed through the process instead of "clicking Next," as stated in the following procedure.

**Note:** Before you uninstall, we recommend copying your agent configuration settings to have as a backup.

**Follow these steps:**

1. Log in as a root user.
2. Stop the web server.
3. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
4. Open a shell and navigate to the directory where the CA Single Sign-On agent is installed:

   ```
   web_agent_home/install_config_info/ca-wa-uninstall
   ```

5. If necessary, verify that you have execute permissions on the uninstallation program by entering the following command:

   ```
   chmod +x installation_media
   ```

   - ***installation_media***
     Specifies the Policy Server installer executable.

6. Perform one of the following procedures:

   - If you installed the agent using GUI mode, enter the following command from a console window:

     ```
     ./ca-wa-uninstall.sh
     ```

     The uninstallation program starts.

   - If you installed the agent using console mode, use the Configuration Wizard to unconfigure the agent and then delete the *web_agent_home* directory manually.

7. Read the information in the dialog to confirm the removal of the CA Single Sign-On agent, then click Uninstall. The CA Single Sign-On agent is removed from the system.
8. Click Done to exit the uninstallation program.
9. (Optional) Remove the lines from the httpd.conf file that the Configuration Wizard added.
10. Change to your home directory (the current directory has been deleted).
11. Restart the web servers.

# Web Agent for Domino

The following sections detail how to install and configure an agent on a Domino web server.

# Hardware Requirements for a Domino Agent

- **Windows**
  agents operating on Windows require the following hardware:
  - CPU: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in the temporary location.
- **UNIX**
  Agents operating on UNIX require the following hardware:
  - CPU:
    - Solaris operating environment: SPARC
    - Red Hat operating environment: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in /temp.

**Note**: Daily operation of the agent requires 10 MB of free disk space in /tmp. The agent creates files and named pipes under /tmp. The path to which these files and pipes are created cannot be changed.

# How to Prepare for a Web Agent Installation on Domino

This content describes how to prepare for Web Agent installation on Domino.


- Domino Server Preparations for Windows Operating Environments (see page 559)
- Domino Server Preparations for UNIX Operating Environments (see page 559)
- Domino Server Preparations for Linux Operating Environments (see page 560)

## Domino Server Preparations for Windows Operating Environments

Domino servers running on Windows operating environments require the Microsoft Visual C++ 2005 Redistributable Package. Download and install the appropriate package for your operating environment (x86 or x64) from the Microsoft web site (http://www.microsoft.com/).

## Domino Server Preparations for UNIX Operating Environments

Domino servers running on UNIX operating environments require the following preparations before installing an agent:

1. Set the display variable (see page ).
2. Verify that the appropriate Solaris patches (see page 559) have been installed for your operating environment.

### Set the DISPLAY For Agent Installations on UNIX

If you are installing the agent on a UNIX system from a remote terminal, such as a Telnet or Exceed terminal, be sure the DISPLAY variable is set for the local system. For example, if your machine is 111.11.1.12, set the variable as follows:

DISPLAY=111.11.1.12:0.0

export DISPLAY

**Note:** You can also install the agent using the console mode installation, which does not require the X window display mode.

### Required Solaris Patches

Before installing a agent on a Solaris computer, install the following patches:

- **Solaris 9**
  Requires patch 111711-16.
- **Solaris 10**
  Requires patch 119963-08.

You can verify installed patch versions by logging in as the root user and executing the following command:

```
showrev -p | grep patch_id
```

To locate Solaris patches, go to the Oracle Solution Center.

# Domino Server Preparations for Linux Operating Environments

Before installing an agent on a Domino servers running on Linux operating environments, verify that the required Linux software packages and libraries are installed.

## Required Linux Software Packages

The following software packages are required to install Web Agents on 64-bit Linux systems:

- Binutils 2.17
- GCC 4.1.0

## Required Linux Libraries

Certain library files are required for components operating on Linux systems. Failure to install the correct libraries can cause the following error:

```
java.lang.UnsatisfiedLinkError
```

If you are installing, configuring, or upgrading a Linux version of this component, the following packages are required on the host system:

**Red Hat 5.x:**

- compat - gcc-34-c++-3.4.6-*patch_version*.I386
- libstdc++-4.x.x-x.el5.i686.rpm
- libidn.so.11.rpm

**Red Hat 6.x:**

- libstdc++-4.x.x-x.el6.i686.rpm
- libidn-1.18-2.el6.i686
- libXext.i686.rpm
- libXrender.i686.rpm
- libXtst.i686.rpm
- libidn.so.11.rpm

**Additionally, for Red Hat 6.x (64-bit):**

All the RPM packages that are required for 64-bit Red Hat 6.x are *32-bit* packages.

- libXau-1.0.5-1.el6.i686.rpm
- libxcb-1.5-1.el6.i686.rpm

- compat-db42-4.2.52-15.el6.i686.rpm
- compat-db43-4.3.29-15.el6.i686.rpm
- libX11-1.3-2.el6.i686.rpm
- libXrender-0.9.5-1.el6.i686.rpm
- libexpat.so.1 (provided by expat-2.0.1-11.el6_2.i686.rpm)
- libfreetype.so.6 (provided by freetype-2.3.11-6.el6_2.9.i686.rpm)
- libfontconfig.so.1 (provided by fontconfig-2.8.0-3.el6.i686.rpm)
- libICE-1.0.6-1.el6.i686.rpm
- libuuid-2.17.2-12.7.el6.i686.rpm
- libSM-1.1.0-7.1.el6.i686.rpm
- libXext-1.1-3.el6.i686.rpm
- compat-libstdc++-33-3.2.3-69.el6.i686.rpm
- compat-db-4.6.21-15.el6.i686.rpm
- libXi-1.3-3.el6.i686.rpm
- libXtst-1.0.99.2-3.el6.i686.rpm
- libXft-2.1.13-4.1.el6.i686.rpm
- libXt-1.0.7-1.el6.i686.rpm
- libXp-1.0.0-15.1.el6.i686.rpm
- libstdc++.i686.rpm
- compat-libtermcap.rpm
- libidn.i686.rpm

# Install and Configure Domino Agents on Windows

This section contains the following topics:

# How to Install and Configure an Agent for Domino on Windows

**Contents**



## Gather the Information for the Installation Program

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path

## Run the Installation Program on Windows

The installation program for the agent installs the agent on one computer at a time using the Windows operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation programs also create a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy the Web Agent installation executable file to a temporary directory on your web server.
2. Do *one* of the following steps:
   - For wizard-based installations, right-click the installation executable file, and then select Run as Administrator.
   - For console-based installations, open a command line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```
3. Use the information that you gathered previously to complete the installation.

## Gather the Information that the Configuration Program Requires on Windows

Gather the following information before running the configuration program for the agent on your Domino server:

- **Register Host**
  Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.
  **Default:**Yes
  **Limits:** Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This user account requires privileges to register trusted hosts.

- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.

- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.

- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

- **FIPS Migration/AES Migration**
  Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, the environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

- **FIPS Only/AES Only**
  Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.
  **Default**: FIPS Compatibility/AES Compatibility
  FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

  **Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for the Agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.

- **iNotes File**
  Specifies the location of the iNotes file for a Web Agent running on a Domino web server.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Add the Domino Web Agent DLL (Windows)

To make the Domino Web Agent operate properly, add the DOMINOWebAgent.dll file to the filter DLLs. The Web Agent DLL must be the first DLL in the list.

**Follow these steps:**

1. Open Lotus Notes.
2. Select File, Database, Open.
3. In the Server field, select the Domino Server where you installed the Web Agent.
4. In the Database scroll box, select the address book of the server.
   Verify that names.nsf appears in the Filename field.
5. Click Open.
6. In the left pane, expand the Server folder and double-click the All Server Documents icon.
7. Select your server and click Edit Server.
8. Select the Internet Protocols tab.
9. In the DSAPI section of the window, find the DSAPI filter file names field and enter the full path to the Domino Web Agent DLL. Verify that the Domino Web Agent DLL appears first in the list. The default location of this DLL file is shown in the following example:
   *web_agent_home*\bin\DOMINOWebAgent.dll

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On agent is installed on your web server.
     **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32
10. Click Save and Close.
11. Restart the web server. In some situations, a reboot could possibly be necessary.

## Run the Web Agent Configuration Program on Windows

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console modes once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:
   *web_agent_home*\install_config_info

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On agent is installed on your web server.
     **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent

**Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
**Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32

2. Use *one* of the following configuration methods:
   - For a GUI-based configuration, go to Step 3.
   - For a console-based configuration, go to Step 5.
3. Right-click the following executable, and then select Run as Administrator:

   ```
   ca-wa-config.exe
   ```
4. Go to Step 8.
5. Open a Command Prompt window with Administrator privileges.
6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```
7. Go to Step 8.
8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## Configure the CGI Directory and CGI URL Path Settings on Windows Operating Environments (Optional)

To configure appropriate cgi-bin (ScriptAlias) settings for the Domino Web Agent, navigate to Internet Protocols tab for your server configuration in the Domino Administrator and configure the following settings:

- CGI directory: domino\html\cgi-bin
- CGI URL path: /cgi-bin

## Configure Alias Settings to Enable HTML Forms Authentication Schemes (Optional)

To configure the Domino Web Agent to support HTML Forms authentication schemes perform the following tasks:

1. Create a subdirectory named "siteminderagent" in the Domino document root (\domino\html\) directory.
2. Copy all the subdirectories of *agent_home*\samples to the siteminderagent directory you created in Step 1.
   - **agent_home**
     Specifies the CA Single Sign-on Web Agent installation path.

3. To support X.509 Client Certificate and HTML Forms authentication schemes, additionally create a directory named "certoptional" in the siteminderagent directory you created in Step 1 and also copy all the subdirectories of *agent_home*\samples into it.

## Run a Silent Installation and Configuration for Domino Agents on Windows

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The Web Agent Installation wizard.
   b. The Web Agent Configuration wizard.
2. Locate the following file on your first web server:
   *web_agent_home*\install_config_info\ca-wa-installer.properties

   **Note**: If the path contains spaces, surround it with quotes.

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On agent is installed on your web server.
     **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32
3. Perform each of the following steps on the other web servers in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.
   b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:

- The Web Agent Installation executable file.
- ca-wa-installer properties file.

c. Open a Command Prompt window with Administrative privileges in the temporary directory.

d. Run the following command:

```
agent_executable -f properties_file -i silent
```

The agent is installed and configured on the subsequent server silently.

e. (Optional) Delete the temporary directory from your subsequent web server.

4. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Install and Configure Domino Agents on UNIX Linux

This section contains the following topics:

- How to Install and Configure an Agent for Domino on UNIX Linux (see page 568)
- Run a Silent Installation and Configuration for Domino Agents on UNIX Linux (see page 575)

## How to Install and Configure an Agent for Domino on UNIX Linux

**Contents**



### Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Run the Installation Program on UNIX Linux

The installation program for the CA Single Sign-On agent installs the agent on one computer at a time using the UNIX or Linux operating environments. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy CA Single Sign-On agent installation executable file to a temporary directory on your web server.
2. Log in as a root user.
3. Do *one* of the following steps:
   - For wizard-based installations, run the installation executable file.
   - For console-based installations, open a command-line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```
4. Use the information from your agent Installation worksheet to complete the installation program.

## Gather the Information that the Configuration Program Requires for UNIX or Linux

Gather the following information before running the configuration program for the agent on your Domino server:

- **Register Host**
  Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.
  **Default:**Yes
  **Limits:** Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This user account requires privileges to register trusted hosts.

- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.

- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.

- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, the environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.
    **Default**: FIPS Compatibility/AES Compatibility
    FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

    **Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for the Agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.

- **iNotes File**
  Specifies the location of the iNotes file for a Web Agent running on a Domino web server.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Add the Domino Web Agent Library File (UNIX)

To make the Domino Web Agent operate properly, add the libdominowebagent.so library file to the filter files. The Web Agent library file must be the first file in the list.

**Follow these steps:**

1. Open Lotus Notes.
2. Select File, Database, Open.
3. In the Server field, select the Domino Server where you installed the Web Agent.
4. In the Database scroll box, select the address book of the server.
   Verify that names.nsf appears in the Filename field.
5. Click Open.
   The address book of the server opens.
6. In the left pane, expand the Server folder and double-click the All Server Documents icon.
7. Select your server and click Edit Server.
   The administration console of the Domino server opens.
8. Select the Internet Protocols tab.

9. In the DSAPI section of the window, find the DSAPI filter file names field and enter the full path to the Domino Web Agent file. Verify that the Domino Web Agent file appears first in the list. The default location of the file is shown in the following example:
web_agent_home\bin\libdominowebagent.so

   - **web_agent_home**

     Indicates the directory where the CA Single Sign-On Agent is installed.

     **Default** (UNIX/Linux installations): /opt/ca/webagent

   > **Note:** If the Domino Web Agent is installed on an AIX operating system, the file name of the Domino Web Agent for the DSAPI filter is libdominowebagent.a

10. Click Save and Close.

11. Restart the web server.

## Source the Agent Environment Script on UNIX or Linux

The agent installation program creates an environment script, **ca_wa_env.sh** in the following directory:

`*web_agent_home*/ca_wa_env.sh`

*web_agent_home* indicates the directory where the Agent is installed. The default UNIX/LINUX location for the script is:

`opt/ca/webagent`

For RHEL 7, include the content of the source script in the directory:

`/etc/sysconfig/httpd`

The following is a sample of the modified script in the directory /etc/sysconfig/httpd. Strings in **bold** are in effect and others are commented out.

Note the following:

- Replace any ${*VARIABLE*} with the actual value.
- To determine the values for the variables ${LD_LIBRARY_PATH} and ${PATH}, use the **env** command before you add the script contents.

```
NETE_WA_ROOT=/opt/CA/webagent
export NETE_WA_ROOT
NETE_WA_PATH=/opt/CA/webagent/bin
#NETE_WA_PATH=${NETE_WA_ROOT}/bin
export NETE_WA_PATH
 CAPKIHOME=/opt/CA/webagent/CAPKIexport CAPKIHOME
LD_LIBRARY_PATH=/opt/CA/webagent/bin:/opt/CA/webagent/bin/thirdparty#LD_LIBRARY_PATH=$
{NETE_WA_ROOT}/bin:${NETE_WA_ROOT}/bin/thirdparty:${LD_LIBRARY_PATH}
export LD_LIBRARY_PATH
```

```
PATH=/opt/CA/webagent/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr
/bin:/root/bin
#PATH=/opt/CA/webagent/bin:${PATH}
#PATH=${NETE_WA_PATH}:${PATH}
export PATH
```

For most Apache-based web servers, source this script *before* doing any of the following tasks:

- Running the agent configuration program.
- Starting the web server.

**Note**: If you perform *all* the previous tasks in the *same* shell, only source the script *once*.

For the embedded Apache web server included with RedHat Linux, do *one* of the following tasks:

- Source the script *before* starting the httpd service.
- Source the script in the following file instead of starting it manually each time:

  `/etc/init.d/htppd`

## Set the Library Path Variable on UNIX or Linux

Set the library path variable on UNIX or Linux systems before running the agent configuration program.

The following table lists the library path variables for the various UNIX and Linux operating environments:

| Operating System | Name of Library Path Variable |
|---|---|
| AIX | LIBPATH |
| Linux | LD_LIBRARY_PATH |
| Solaris | LD_LIBRARY_PATH |

Set the value of the library path variable to the *agent_home*/bin directory.

- ***agent_home***
  Indicates the directory where the Agent is installed.

## Run the Web Agent Configuration Program on UNIX Linux

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:
   *web_agent_home/*install_config_info

   - **web_agent_home**
     Indicates the directory where the CA Single Sign-On Agent is installed.
     **Default** (UNIX/Linux installations): /opt/ca/webagent

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, go to Step 3.
   - For a console-based configuration, go to Step 5.

3. Run the following executable file:

   ```
   ca-wa-config.bin
   ```

4. Go to Step 8.

5. Open a Command Prompt window with root privileges.

6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```

7. Go to Step 8.

8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## Configure Alias Settings to Enable HTML Forms Authentication

To configure the Domino Web Agent to support HTML Forms authentication schemes perform the following tasks:

1. Create a subdirectory named "siteminderagent" in the Domino document root (\domino\html\) directory.

2. Copy all the subdirectories of *agent_home*\samples to the siteminderagent directory you created in Step 1.

   - **agent_home**
     Specifies the CA Single Sign-on Web Agent installation path.

3. To support X.509 Client Certificate and HTML Forms authentication schemes, additionally create a directory named "certoptional" in the siteminderagent directory you created in Step 1 and also copy all the subdirectories of *agent_home*\samples into it.

# Run a Silent Installation and Configuration for Domino Agents on UNIX Linux

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.
For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The CA Single Sign-On Web Agent Installation wizard.
   b. The CA Single Sign-On Web Agent Configuration wizard.
2. Locate the following file on your first web server:
   *web_agent_home*/install_config_info/ca-wa-installer.properties

   **Note**: If the path contains spaces, surround it with quotes.

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On agent is installed on your web server.
3. Perform each of the following steps on the other web servers in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.
   b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:
      - The CA Single Sign-On Web Agent Installation executable file.
      - ca-wa-installer properties file.
   c. Open a Command Prompt window with Administrative privileges in the temporary directory.
   d. Run the following command:

```
agent_executable -f properties_file -i silent
```

The agent is installed and configured on the subsequent server silently.

e. (Optional) Delete the temporary directory from your subsequent web server.

4. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Uninstall a Domino Agent from Windows

Before you un–install the CA Single Sign-On Web Agent from a Windows operating environment, consider making backup copies of your registry settings and Web Agent configuration settings. Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

**Follow these steps:**

1. Stop the web server.
2. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
3. Choose *one* of the following procedures:
   - To remove the Web Agent using the wizard, go to Step 4.
   - To remove the Web Agent using the console-based program, go to Step 9.
4. Click Start, Control Panel, Programs and Features.
5. Click CA Single Sign-On Web Agent *version_number*.
6. Click Uninstall/Change.
7. Review the information in the Uninstall CA Single Sign-On Web Agent dialog, then click Uninstall.
8. Wait for the wizard to finish, then go to Step 13.
9. Open a command-line window.
10. Navigate to the following directory.
    *web_agent_home*

- ***web_agent_home***
  Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
  **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
  **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
  **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

11. Run the following command:

    ```
    ca-wa-uninstall.cmd -i console
    ```

12. Wait for the un–installation program to finish, then go to Step 13.

13. Start the web server.

**Important!** Delete the ZeroG registry file from the following location after uninstalling the Web Agent: C:\Program Files\ZeroG Registry\com.zerog.registry.xml

# Uninstall a Domino Agent from UNIX

These instructions are for GUI and Console Mode removal of an agent.
Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

**Note:** Removing a Web Agent from a 64-bit SuSE Linux 10 system requires additional preparations.

The steps for the two modes are the same, with these exceptions for Console Mode:

- Select an option by entering a corresponding number.
- Press Enter after each step to proceed through the process. The prompts guide you through the process.

> **Note:** Before you uninstall, we recommend copying your Web Agent configuration settings to have as a backup.

**Follow these steps:**

1. Stop the web server.
2. Log in to the UNIX system.
3. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
4. Navigate to the directory where the Web Agent is installed: *web_agent_home* /install_config_info/ca-wa-uninstall
5. If necessary, verify that you have execute permissions on the uninstallation program by entering chmod +x uninstall.
6. From a console window, enter one of the following commands:
   - GUI mode: ./uninstall
   - Console mode: ./uninstall -i console

   The uninstallation program starts.
7. Read the information in the dialog to confirm the removal of the Web Agent, then click Uninstall. The Web Agent is removed from the system.
8. Click Done to exit the uninstallation program.
9. Change to your home directory (the current directory has been deleted).
10. Restart the web servers.

# Web Agent for IIS

The following sections detail how to install and configure an agent on an IIS web server.

## Hardware Requirements for an IIS Agent

- **Windows**
  agents operating on Windows require the following hardware:
  - CPU: x86 or x64
  - Memory: 2-GB system RAM.

- Available disk space:
  - 2-GB free disk space in the installation location.
  - .5-GB free disk space in the temporary location.

# Combined Functions in New Agent for Internet Information Services (IIS) Web Servers

This product combines all functions for Internet Information Services (IIS) into one agent.

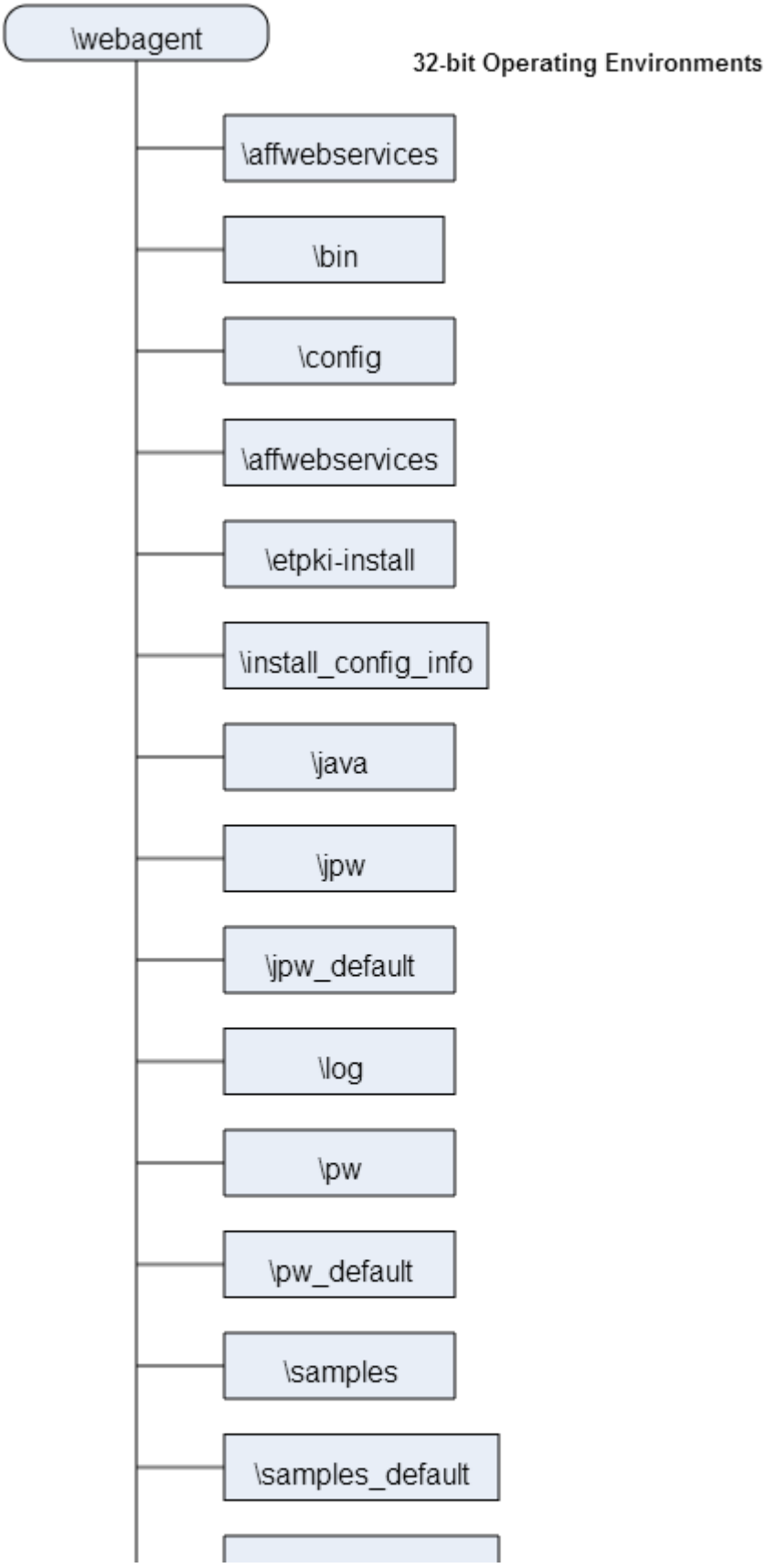
A Web Agent for IIS implemented as an ISAPI plug-in and a native HTTP module that supports the following functions:

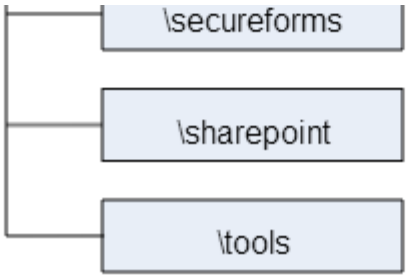
- Application pools using Integrated or Classic pipeline mode.
- Application pools that are configured with the Enable 32-bit applications option.
- The optional IIS Application Request Routing (ARR) feature.
- Supported with IIS 7.x and higher versions, including IIS clusters and shared configuration deployments.

# Multiple Agent for IIS Directory Structures

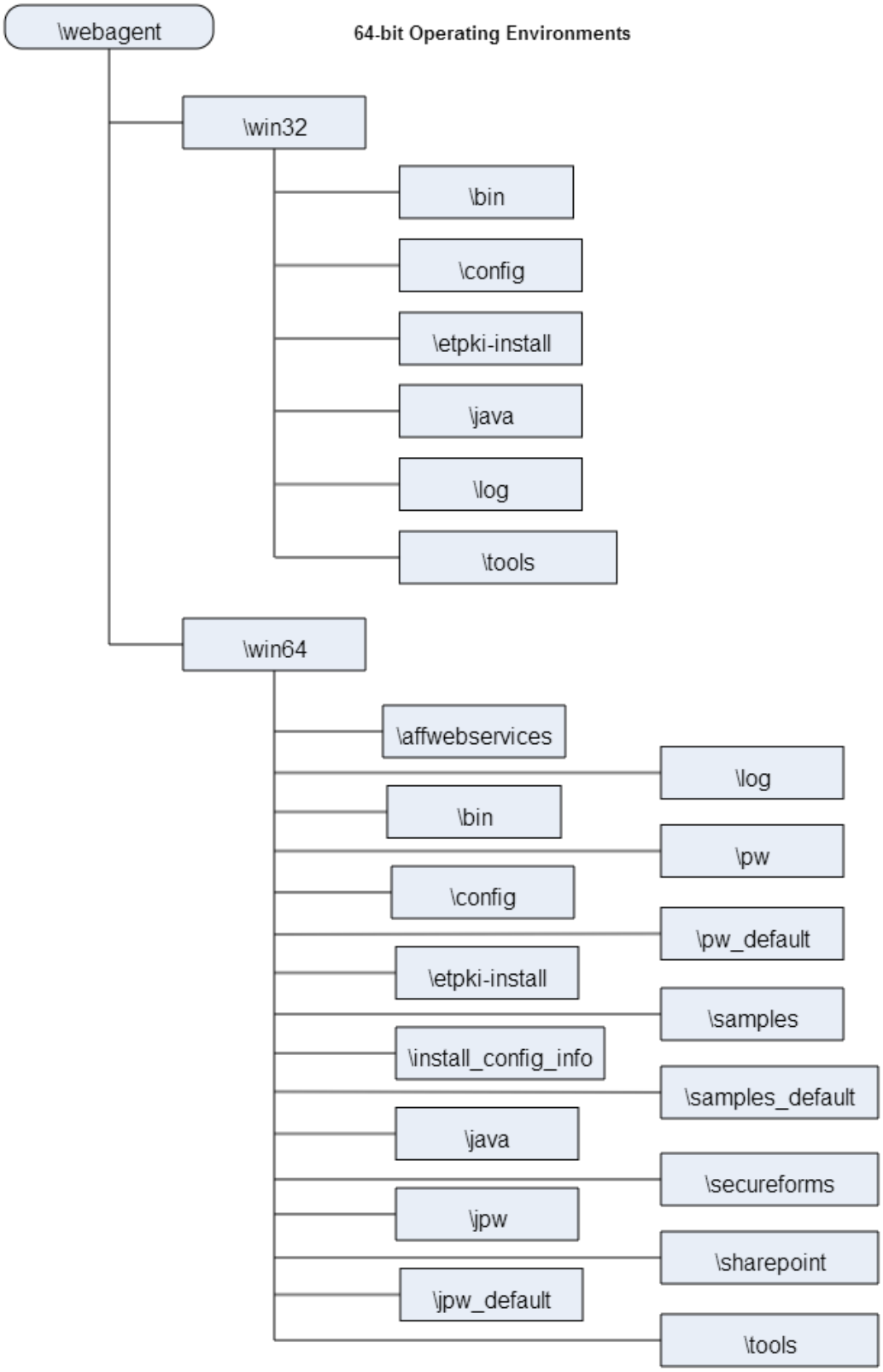
The directory structure added to your IIS web server for your Agent files varies according to the operating environment of your IIS web server. The following directory structures exist:

- CA Single Sign-On Agents for IIS use the 32-bit directory structure shown in the following illustration:


\webagent
32-bit Operating Environments
\affwebservices
\bin
\config
\affwebservices
\etpki-install
\install_config_info
\java
\jpw
\jpw_default
\log
\pw
\pw_default
\samples
\samples_default




- CA Single Sign-On Agents for IIS installed on 64-bit operating environments use the directory structure shown in the following illustration:


\webagent
64-bit Operating Environments
\win32
\bin
\config
\etpki-install
\java
\log
\tools
\win64
\affwebservices
\log
\bin
\pw
\config
\pw_default
\etpki-install
\samples
\install_config_info
\samples_default
\java
\secureforms
\jpw
\sharepoint
\jpw_default
\tools

# How to Prepare for an Agent for IIS Installation

**Contents**



## Verify that you have an Account with Administrative Privileges

To install or configure a CA Single Sign-On Web Agent or CA Single Sign-On Agent for IIS on an IIS web server, you need an account with Administrator privileges.

For Windows 2008 systems, do one of the following actions to install or configure a CA Single Sign-On Web Agent or CA Single Sign-On Agent for IIS:

- If you are using Windows Explorer, right-click the .exe file. Then select Run as Administrator.
- If you are using a command line, open a new console window with administrative privileges. Then run the command that you want.

**Note**: For more information about installing or configuring CA Single Sign-On Web Agents or CA Single Sign-On Agents for IIS on Windows 2008 systems, see the Web Agent Release Notes.

## Verify that the IIS Role and Role Services are Installed

The IIS (web server) role is *not* enabled by default. Verify that the IIS role is installed and enabled on each Windows system, before installing the Agent for IIS.

**Follow these steps:**

1. Click Start, All Programs, Administrative Tools, Server Manager.
2. Verify that IIS appears in the Roles list.
3. If the Web Server (IIS) role is not shown, add it using the Add Roles wizard. If you decide to use the ISAPI-filter functions of the Agent for IIS, add the following role services too:
   - CGI
   - ISAPI Extensions

- ISAPI Filters
- IIS Management Console
- Windows Authentication (for the CA Single Sign-On Windows Authentication Scheme)

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Verify that the Windows IIS Web Server has the Latest Service Packs and Updates

We recommend using Windows Update to verify that your Windows operating environment contains the latest Service Packs and updates, before installing a CA Single Sign-On Agent for IIS.

## Review the Policy Server Prerequisites for Agent for IIS Installations

Your Agent for IIS needs the following information about the Policy Servers to which it connects:

- The IP addresses of the Policy Servers
- Certain CA Single Sign-On object names in the Policy Server

The Administrative UI creates these objects in the Policy Server. We recommend creating them before installing your agent to avoid going between your web server and the Administrative UI interfaces later.

Agents for IIS require the names of the following CA Single Sign-On objects stored the Policy Server:

- **Host Configuration Object**
  Contains the settings that the agent uses for subsequent connections to a Policy Server following the initial connection that the agent made.

- **Admin User Name**
  Identifies the name of a CA Single Sign-On user with the following privileges:

  - Administrative privileges

  - Trusted host registration privileges

- **Admin Password**
  Identifies a password that is associated with the Admin User Name in the CA Single Sign-On Policy Server.

- **AgentName**
  Defines the identity of the Web Agent. This identity establishes a mapping between the name and the IP address of each web server instance hosting an Agent.
  When no matching value exists, the agent uses the value of from the DefaultAgentName parameter instead.
  **Note**: This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add the parameter name and a value to separate lines in the file.
  **Default**: No default
  **Limit**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
  **Limits**: Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. Not case-sensitive. For example, the names MyAgent and myagent are treated the same.
  **Example**: myagent1,192.168.0.0 (IPV4)
  **Example:** myagent2, 2001:DB8::/32 (IPV6)
  **Example**: myagent, www.example.com (http://www.example.com)

# Install and Configure an IIS Agent

This section contains the following topics:

- Install and Configure an Agent for IIS (see page 585)
- Run a Silent Installation and Configuration on an IIS Agent (see page 594)
- How to Configure Certain Settings for the Agent for IIS Manually (see page 603)

## Install and Configure an Agent for IIS

**Contents**

- IIS Web Server Shared Configuration and the Agent for IIS (see page 586)

**Note:** The following information is applicable to IIS 7.x, IIS 8.x, and IIS 10.

## IIS Web Server Shared Configuration and the Agent for IIS

IIS web servers support shared configurations that streamline the configuration process for an IIS a server farm.

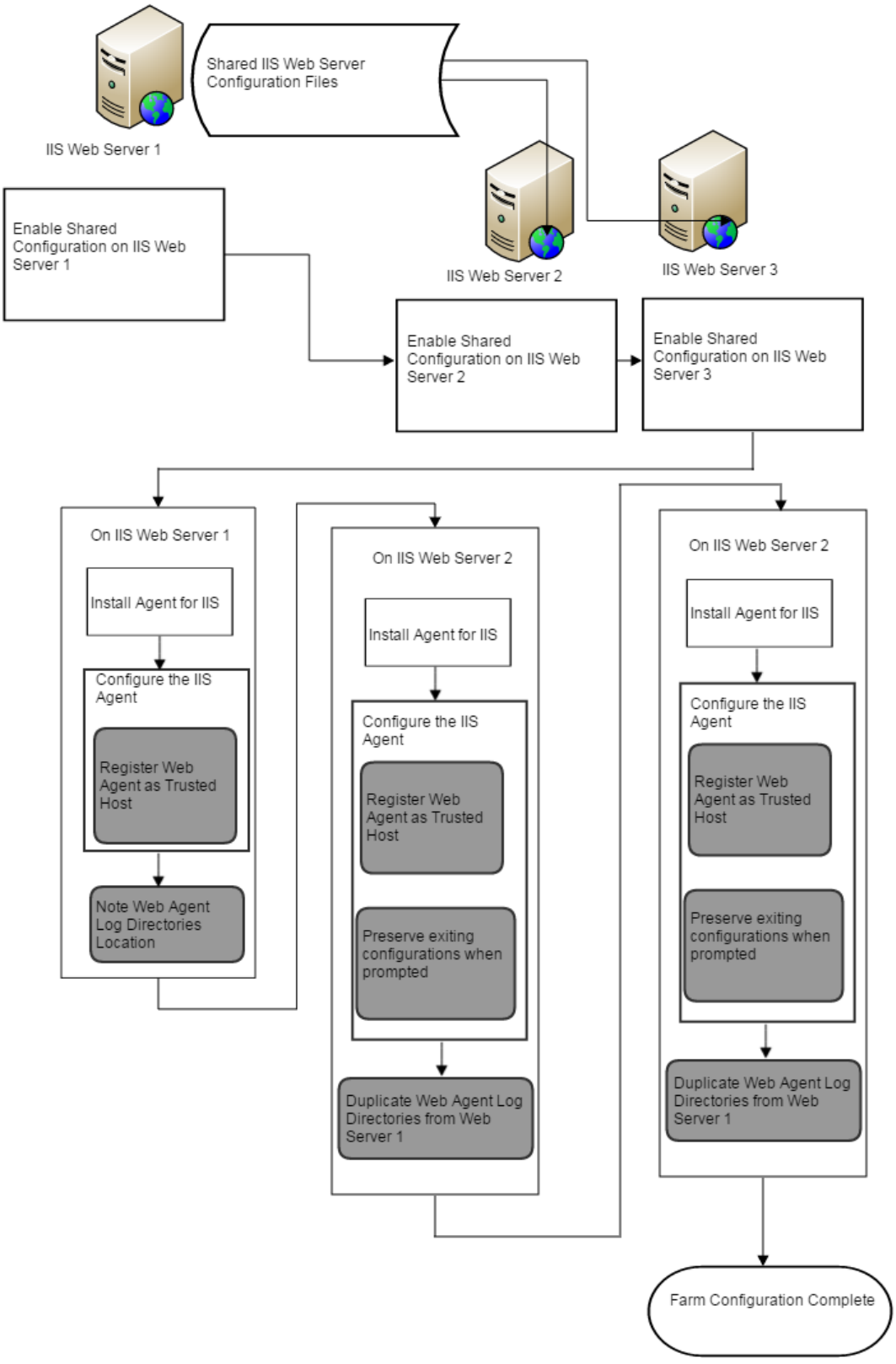
The Agent for IIS can protect resources on IIS server farms that use the shared configuration feature of IIS Web Server.

**Note**: This feature works *only* with the Agent for IIS 7.x and higher versions. Older versions of the Web Agent do *not* support this feature.

IIS web server uses network shares to propagate the configuration information across the server farm. The Agent for IIS, however, *cannot* operate on network shares. Using an Agent for IIS on an IIS server farm involves several separate procedures.

For example, suppose that you have three IIS web servers, with all of them using a shared configuration. Web server number one is your primary web server, which contains the configuration information for the farm. Web servers 2 and 3 are nodes that are connect to the network share on web server 1 to read the configuration information.

The entire installation and configuration process for using the Agent for IIS on all three IIS web servers is described in the following illustration:


IIS Web Server 1
Shared IIS Web Server Configuration Files
IIS Web Server 2
IIS Web Server 3
Enable Shared Configuration on IIS Web Server 1
Enable Shared Configuration on IIS Web Server 2
Enable Shared Configuration on IIS Web Server 3
On IIS Web Server 1
Install Agent for IIS
Configure the IIS Agent
Register Web Agent as Trusted Host
Note Web Agent Log Directories Location
On IIS Web Server 2
Install Agent for IIS
Configure the IIS Agent
Register Web Agent as Trusted Host
Preserve exiting configurations when prompted
Duplicate Web Agent Log Directories from Web Server 1
On IIS Web Server 2
Install Agent for IIS
Configure the IIS Agent
Register Web Agent as Trusted Host
Preserve exiting configurations when prompted
Duplicate Web Agent Log Directories from Web Server 1
Farm Configuration Complete

## How Web Agent Logs and Trace Logs Work with IIS Web Server Shared Configuration

For Agents for IIS running on an IIS server farm, create duplicate log and trace file directories on each node if all the following conditions are true:

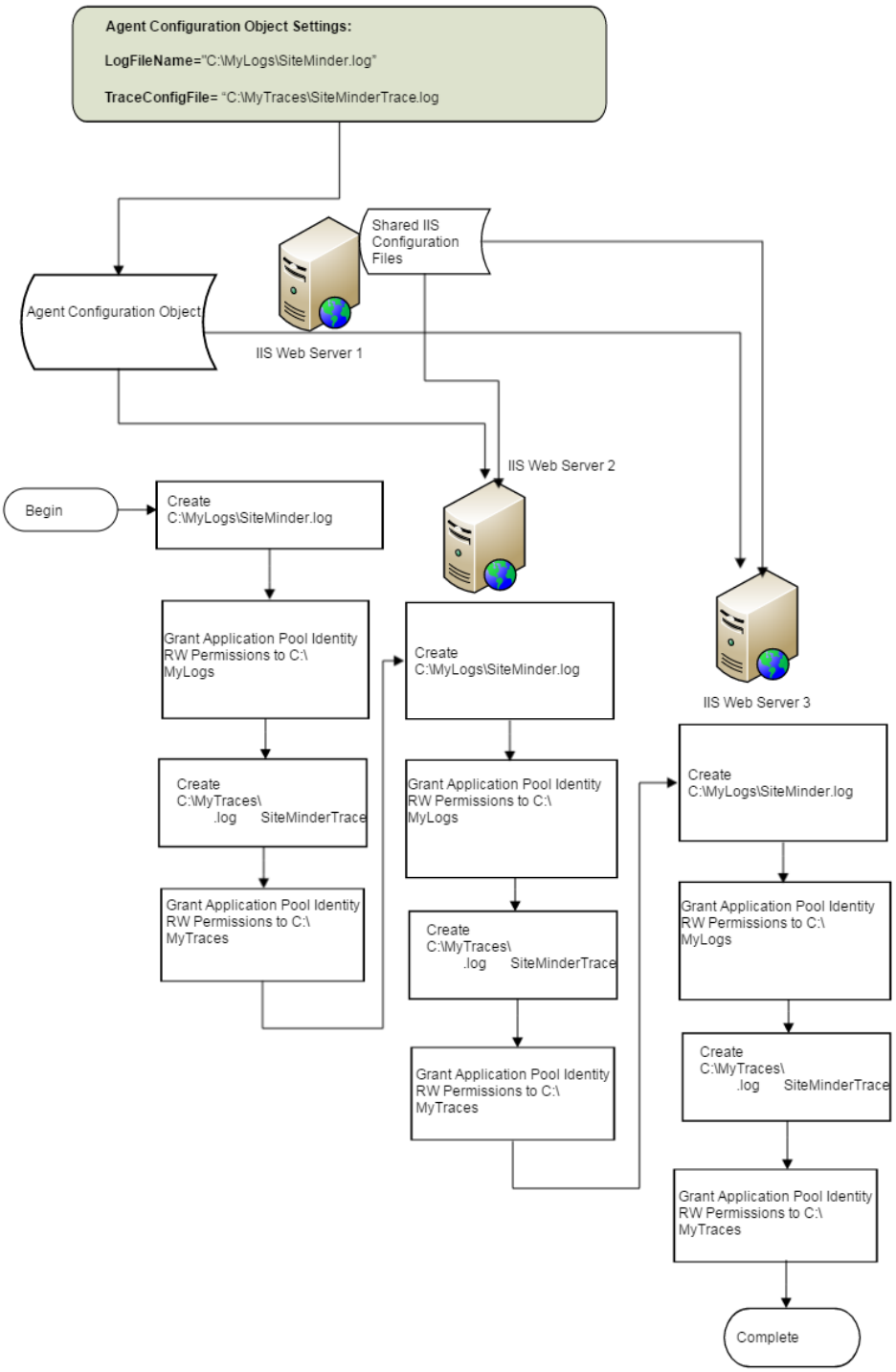
- Your Agent for IIS log and trace log directories are specified in an Agent Configuration Object on the Policy Server (*not* in a local configuration file).
- Any of the Agents for IIS in your IIS web servers in the server farm share the same Agent Configuration object
- Your Agent for IIS log file and trace log directories that are specified in the shared Agent Configuration Object are *different from* the following default settings:
  - *web_agent_home*\win32\log (for Windows IIS web server 32-bit)
  - *web_agent_home*\win64\log (Windows IIS web server 64-bit)

If all the previous conditions exist in your server farm, use the following process to enable your Web Agent logs and trace logs:

1. Create a custom log directory on the IIS web server that contains the shared configuration for the farm.
2. Grant the application pool identities associated with your protected resources the following permissions to the custom directory on the previous IIS web server.
   - Read
   - Write
3. Create the same custom log directory on a IIS web server node in the farm.
4. Grant the application pool identities associated with your protected resources the following permissions to the custom directory on the IIS web server node in the farm.
   - Read
   - Write
5. Repeat steps 3 and 4 on all other nodes in your server farm.

For example, suppose that you have three IIS web servers, with all of them using a shared configuration. Web server number one is your primary web server, which contains the configuration information for the farm. Web servers 2 and 3 are nodes that connect to the network share on web server one to read the configuration information.

The entire process for configuring these logs is described in the following illustration:




## Gather Information for the Agent Installation Program

Before running the installation program for the Agent for IIS on the Windows operating environment, gather the following information about your web server:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Value**: The path requires the name "webagent" as the last directory in the path.

- **Shortcut Location**
  Specifies the location in your Start menu for the shortcut for the Web Agent Configuration wizard.

## Run the Installation Program on Windows

The installation program for the agent installs the agent on one computer at a time using the Windows operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation programs also create a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object, and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy the Web Agent installation executable file to a temporary directory on your web server.
2. Do one of the following steps:
   - For wizard-based installations, right-click the installation executable file, and then select Run as Administrator.
   - For console-based installations, open a command line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```
3. Use the information that you gathered previously to complete the installation.

## Gather Information for the Agent Configuration Program for IIS Web Servers

Before configuring an Agent on an IIS web server, gather the following information about your environment.

- **Host Registration**
  Indicates whether you want to register this agent as a trusted host with a Policy Server. Only one registration per agent is necessary. If you are installing the Agent for IIS on an IIS server farm, register all IIS agents in the farm as trusted hosts.
  **Options**: Yes, No

- **Admin User Name**
  Specifies the name of a user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Admin Password**
  Specifies the password that is associated with the user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Confirm Admin Password**
  Confirms the password that is associated with the user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Enable Shared Secret Rollover**
  Indicates whether the Policy Server generates a new shared secret when the agent is registered as a trusted host.

- **Trusted Host Name**
  Specifies a unique name for the host you are registering. After registration, this name appears in the list of Trusted Hosts in the Administrative UI. When configuring an Agent for IIS on an IIS web server farm, specify a *unique* name for *each* IIS server node on the farm. For example, if your farm uses six servers, specify six unique names.

- **Host Configuration Object**
  Indicates the name of the Host Configuration Object that exists on the Policy Server.

- **IP Address**
  Specifies the IP addresses of any Policy Servers to which the agent connects. Add a port number if you are *not* using the default port for the authentication server. Non-default ports are used for all three Policy Server connections (authentication, authorization, accounting).
  **Default**: (authentication port) 44442
  **Example:** (IPv4) 127.0.0.1,55555
  **Example:** (IPv6) [2001:DB8::/32][:55555]

  **Note:** If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.

- **FIPS Mode Setting**
  FIPS is a United States government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES). Select *one* of the following options:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data. The algorithm must be compatible with the previous versions in use. If your organization does not require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, the Policy Server continues to use existing encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

- **FIPS Only/AES Only**
  Uses only FIPS-compliant algorithms to encrypt sensitive data in the environment. This setting does not interoperate with, nor is backwards-compatible with, previous versions of the product.
  **Default**: FIPS Compatibility/AES Compatibility

> Important! Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored. On Windows 64-bit operating environments, the configuration program creates two separate files. One file supports 64-bit applications, and the other file supports 32-bit applications running on the same web server.
  **Default**: (Windows IIS web server 32-bit) *web_agent_home*\win32\bin\IIS
  **Default**: (Windows IIS web server 64-bit) *web_agent_home*\win64\bin\IIS

- **Virtual Sites**
  Lists the web sites on the IIS web server that you can protect with the Agent.

> **Important!** Do not configure and unconfigure virtual sites simultaneously. Run the wizard once to configure the sites you want, and then run the wizard again to unconfigure the sites you want.

- **Overwrite, Preserve, Unconfigure**
  Appears when the Agent configuration wizard detects *one* of the following situations:

  - IIS websites that the Agent already protects on a stand-alone IIS web server.
  - IIS websites that Agent protects on an IIS server farm using shared configuration.

  Select *one* of the following options:

  - **Overwrite**
    Replaces the previous configuration of the Agent with the current configuration.
  - **Preserve**
    Keeps the existing configuration of your Agent. No changes are made to this web server instance. Select this setting for each web server node if you are configuring the Agent for IIS on an IIS server farm.

- **Unconfigure**
  Removes the existing configuration of an Agent from the web server. Any resources are left unprotected.
  **Default**: Preserve

- **Agent Configuration Object Name**
  Specifies the name of an Agent Configuration Object (ACO) already defined on the Policy Server. IIS web servers in a server farm using shared configuration support sharing a single ACO name with all IIS servers in the farm.
  **Default**: AgentObj

## Run the Web Agent Configuration Wizard

After gathering the information for your Agent Configuration worksheet, run the Agent Configuration wizard. The configuration wizard creates a runtime instance of the Agent for IIS on your IIS web server.

Running the configuration wizard once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment and settings.

**Note**: The configuration wizard for this version of the Agent for IIS does *not* support console mode.

**Follow these steps:**

1. Click Start, All Programs, CA, Single Sign-On.
   A shortcut to the Web Agent Configuration wizard appears.

2. Right-click the shortcut, and then select Run as administrator.

   **Important!** On Windows Server, if User Account Control (UAC) is enabled open the shortcut with Administrator permissions. Use Administrator permissions even if you are logged in to the system as an Administrator. For more information, see the Release Notes for your prod component.

   The Web Agent Configuration wizard starts.

3. Complete the wizard.

4. After the Web Agent is configured, restart the IIS server and the Web Agent to ensure that all current connections are cleared and all the website changes are applied.
   To restart the server, do one of the following:

   - Enter the following command at the command line:

     ```
     c:\> iisreset
     ```

- Access the Internet Information Services (IIS) Manager and click on Actions -> All Tasks -> Restart IIS.

## Verify that the ISAPI Filter is First in the List When Using Classic Pipeline Mode

Applications running in classic pipeline mode require that the ISAPI filter appears first in the list of ISAPI filters. Verify the position of the ISAPI filter in the list of ISAPI filters on your IIS web server before continuing.

**Follow these steps:**

1. Open IIS Manager using the following steps:
   a. Click Start, Control Panel.
   The control panel opens.
   b. Click Administrative Tools, Internet Information Services (IIS) Manager.
   IIS Manager opens.
2. Verify that the ISAPI filter is first in the list using the following steps:
   a. From IIS Manager, expand the following items:
      - Your web server
      - Sites
      - Default Web Site
   b. Double-click the Handler Mappings icon.
   c. Click view ordered list.
   d. Verify that the following ISAPI filter appears in the top of the list:
      `handler-wa`
3. If the ISAPI filter from Step 2d does *not* appear first in the list, do the following steps:
   a. Click the handler-wa ISAPI filter.
   b. Click the Move up arrow until the ISAPI filter appears first in the list.

The ISAPI filter appears first in the list.

# Run a Silent Installation and Configuration on an IIS Agent

**Contents**

- Remove a Web Agent Configuration from an IIS Web Server Silently (see page 597)
- Remove CA Single Sign-On Protection from Some Virtual Sites on IIS Web Servers Silently (see page 599)
- Add CA Single Sign-On Protection to Additional Virtual Sites on IIS Web Servers Silently (see page 601)

The unattended or silent installation can help you automate the installation and configuration process, thus saving time for a large CA Single Sign-On environment that uses many agents with identical settings.

For example, the Agents in your environment use the same web server version, installation directory, Agent Configuration Object, and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

The unattended installation uses the ca-wa-installer.properties file to propagate the Web Agent installation setup to all Agents in your network. Define the installation parameters in the properties file, copy the properties file, and the Web Agent file to a web server in your network.

**Note**: The default parameters and paths in the file reflect the information that you entered during the initial Web Agent installation.

## Prepare for an Unattended Installation

Perform the following steps to prepare for an unattended installation:

1. Run an initial installation of the Web Agent.
2. Locate the following file on your first IIS web server:

   *web_agent_home*\install_config_info\ca-wa-installer.properties

3. Open the ca-wa-installer.properties file and modify the following parameters:

- USER_INSTALL_DIR
  Defines the Web Agent installed location. Enter the full path to the installation directory.
- (Windows only) USER_SHORTCUTS
  Defines the location to install the Web Agent Configuration Wizard. Enter the path.
- (Windows only) USER_REQUESTED_RESTART
  Specifies if the installation program requires a reboot. Set to YES to allow the reboot.

4. Save the file.

## Run an Unattended Installation

Use the ca-wa-installer.properties file to perform an unattended installation of subsequent Web Agents.
**Follow these steps:**

1. Copy the following files to a local directory from the system where you installed the Web Agent:

a. ca-wa-version-win32.exe

b. ca-wa-installer.properties file from *web_agent_home*\install_config_info

2. Open a command window and navigate to the directory where you copied the files.
3. Run the following command from the directory location:

   ```
   ca-wa-<version>-win32.exe -f ca-wa-installer.properties -i silent
   ```

4. Return to the command prompt when the installation is complete.
5. Verify that the installation is successful in the CA_SiteMinder_Web_Agent_version_InstallLog.log file.
   **Default location**:

   *web_agent_home*\install_config_info directory

6. Register the trusted host and configure the Web Agent.

**Note:** If you are configuring an Agent on an IIS 6.0 server, do not configure the Agent immediately after installation. You must perform additional procedures before you configure the Agent.

## Prepare for an Unattended Configuration

Perform the following steps to prepare for an unattended configuration:

1. Run an initial (attended) Web Agent installation
2. Run an initial (attended) Web Agent configuration
3. Modifiy the ca-wa-installer.properties file to run subsequent unattended Web Agent configurations
4. Run an installation (attended or unattended) on the system where you want to run the unattended configuration.

## Run an Unattended Configuration

Use the ca-wa-installer.properties file to perform an unattended installation of subsequent Web Agents.

**Follow these steps:**

1. Copy the ca-wa-installer.properties file from a Web Agent installed system to a local directory from the following location:

   ```
   web_agent_home/install_config_info
   ```

2. Open a console window and navigate to *web_agent_home*/install_config_info.
3. Run the following command from the directory location:

   **Windows**:

```
ca-wa-config.exe -f ca-wa-installer.properties -i silent
```

**UNIX**:

```
ca-wa-config.bin -f ca-wa-installer.properties -i silent
```

4. Return to the command prompt when you complete the configuration.

5. Verify that the installation is successful in the CA_SiteMinder_Web_Agent_version_InstallLog.log file.
   **Default location**:

   *web_agent_home*\install_config_info directory

## Remove a Web Agent Configuration from an IIS Web Server Silently

To remove the CA Single Sign-On protection from all the websites on an IIS web server without the Web Agent Configuration wizard, use silent or unattended mode. This mode requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   web_agent_home\install_config_info\ca-wa-installer.properties

   > **Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
     **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

2. Perform each of the following steps on the IIS web servers to which you want to remove protection from virtual sites:

   > **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Open the following directory on an IIS web server node.
      *web_agent_home*\install_config_info

b. Copy the CA Single Sign-On ca-wa-installer.properties file from your first IIS web server (from Step 1) to the install_config_info directory on your IIS web server node.

c. Open the CA Single Sign-On ca-wa-installer.properties file with a text editor.

d. Locate the following parameter:

- **UNCONFIGURE_SITES=**
  Specifies the names of IIS 7.x web sites from which to remove CA Single Sign-On protection on an IIS 7.x web server. Verify that these names match the names shown under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.
  Removing the CA Single Sign-On Web Agent configuration from a website leaves its resources *unprotected*.
  For more information, see the comments in the ca-wa-installer.properties file.
  **Example**: Default website, Example4, Example5

e. Enter the names of the websites you want to unconfigure in the previous parameter.

f. Locate the following parameter:

- **CONFIGURE_SITES=**
  Specifies the names of IIS 7.x web sites to protect on an IIS 7.x web server. Verify that these names match the names shown under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.
  For more information, see the comments in the ca-wa-installer.properties file.
  **Example**: Default website, Example1, Example2

g. Verify that the previous parameter contains no website names.

h. Open a Command Prompt window with Administrative privileges.

> **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

i. Run the following command:

```
agent_configuration_executable -f properties_file -i silent
```

**Example**: ca-wa-config.exe -f ca-wa-installer.properties -i silent
The websites are unconfigured on the node automatically.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the specified configuration by the settings in your <flilename>-wa-installer.properties file.

## Remove CA Single Sign-On Protection from Some Virtual Sites on IIS Web Servers Silently

If your IIS web server already has a Web Agent for IIS installed, you can remove protection from some virtual websites on the web server. For example, you want to remove protection from only two of the virtual sites that are named Example4 and Example5 from to your IIS server. Modify the ca-wa-installer.properties file to remove the configuration from those two virtual websites while leaving the protection for the other websites unchanged.

If you do not want to run the configuration wizard, or have many IIS web servers in a server farm, use the silent mode.

The CA Single Sign-On Web Agent Configuration program supports a silent or unattended mode that requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   `web_agent_home\install_config_info\ca-wa-installer.properties`

   **Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
     **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

2. Perform each of the following steps on the IIS web servers from which you want to remove the protection of the additional virtual sites:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want

   a. Copy the CA Single Sign-On ca-wa-installer.properties file from your first IIS web server (from Step 1) to the install_config_info directory on your IIS web server node:

   b. Open the CA Single Sign-On ca-wa-installer.properties file with a text editor.

   c. Locate the following parameter:

- **UNCONFIGURE_SITES=**
  Specifies the names of IIS 7.x web sites from which to remove CA Single Sign-On protection on an IIS 7.x web server. Verify that these names match the names shown under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.
  Removing the CA Single Sign-On Web Agent configuration from a website leaves its resources *unprotected*.
  For more information, see the comments in the ca-wa-installer.properties file.
  **Example**: Default website, Example4, Example5

d. Add the names of the web sites from which you want to remove the configuration to the previous parameter. Remove the names of any other sites on the web server that you want to leave unchanged.

e. Locate the following parameter:

- **HOST_REGISTRATION_YES=**
  Specifies if the agent configuration program registers the agent with a Policy Server. Each web server requires only one trusted host registration is required. Set the value of this parameter to 0 if you have previously registered a web server with the Policy Server as a trusted host.
  **Default**: 1 (yes)
  **Limits**: 0 (no registration), 1 (registration)

f. If the IIS web *server* is *already* registered as a trusted host with the CA Single Sign-On Policy Server, set the previous parameter to 0. Otherwise, the configuration program registers the web server as a trusted host.

g. Open a Command Prompt window with Administrative privileges in the temporary directory.

> **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

h. Run the following command:

```
agent_configuration_executable -f properties_file -i silent
```

**Example**: ca-wa-config.exe -f ca-wa-installer.properties -i silent
The CA Single Sign-On configuration is removed from the selected virtual sites on the node automatically.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the specified configuration by the settings in your <flilename>-wa-installer.properties file.

## Add CA Single Sign-On Protection to Additional Virtual Sites on IIS Web Servers Silently

If your IIS web server already has a Web Agent for IIS installed, you can protect any additional virtual websites on the web server. For example, if you add two new virtual that are sites named Example2 and Example3 to your IIS server, you can protect them with CA Single Sign-On.

If you do not want to run configuration wizard, or if you have many IIS web servers in a server farm, use the silent mode.

The CA Single Sign-On Web Agent Configuration program supports a silent or unattended mode that requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   `web_agent_home\install_config_info\ca-wa-installer.properties`

   **Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
     **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

2. Perform each of the following steps on the IIS web servers to which you want to protect the additional virtual sites:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on an IIS web server node.

   b. Copy the following files from your first IIS web server (from Steps 1 and 2) to the temporary directory on your IIS web server node:

      - CA Single Sign-On Web Agent Configuration executable file (ca-wa-config.exe)
      - CA Single Sign-On ca-wa-installer.properties file

c. Open the CA Single Sign-On ca-wa-installer.properties file with a text editor.

d. Locate the following parameter:

- **CONFIGURE_SITES=**
  Specifies the names of IIS 7.x web sites to protect on an IIS 7.x web server. Verify that these names match the names shown under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.
  For more information, see the comments in the ca-wa-installer.properties file.
  **Example**: Default website, Example1, Example2

e. Add the names of the web sites you want to configure to the previous parameter. Remove the names of any other sites on the web server that you want to leave unchanged.

f. Locate the following parameter:

- **HOST_REGISTRATION_YES=**
  Specifies if the agent configuration program registers the agent with a Policy Server. Each web server requires only one trusted host registration is required. Set the value of this parameter to 0 if you have previously registered a web server with the Policy Server as a trusted host.
  **Default**: 1 (yes)
  **Limits**: 0 (no registration), 1 (registration)

g. If the IIS web *server* is *already* registered as a trusted host with the CA Single Sign-On Policy Server, change the value of the previous parameter to 0. Otherwise, the configuration program registers the web server as a trusted host.

h. Open a Command Prompt window with Administrative privileges in the temporary directory.

> **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

i. Run the following command:

```
agent_configuration_executable -f properties_file -i silent
```

**Example**: ca-wa-config.exe -f ca-wa-installer.properties -i silent
The CA Single Sign-On Web Agent for IIS is installed and configured on the node automatically.

j. (Optional) Delete the temporary directory from your web server node.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the specified configuration by the settings in your <flilename>-wa-installer.properties file.

# How to Configure Certain Settings for the Agent for IIS Manually

**Contents**

In some situations, the CA Single Sign-On Agent configuration programs *cannot* add the proper settings to all the IIS web server directories which need them.

Configure the CA Single Sign-On Agent for IIS settings manually in *any* of the following situations:

- Your CA Single Sign-On Agent for IIS log files are *not* stored in the following default directory: *web_agent_home*\log

  - ***web_agent_home***
    Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
    **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
    **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
    **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

  For example, suppose that you store your log files in the C:\My Logs\CA Single Sign-on directory. Grant this directory permissions.

- You use an authentication scheme which requests or requires client certificates.

## Set Permissions Manually for Non-Default Log Locations

If you decide to store your agent log files in a non default directory, grant your application pools permissions to the directory. For example, if you want to store your log files in a directory named C:\MyLogFiles, grant permissions for all your application pool identities to C:\MyLogFiles.

Microsoft provides a command line utility, icacls.exe you can use to set the appropriate permissions. This procedure provides one possible example of a way to set permissions using tools or utilities provided by third-party vendors.

**Important!** CA provides this information only as an example of one possible method of configuring CA Single Sign-On without using the programs and utilities tested and approved by CA. Microsoft provides the icacls.exe command as part of the Windows operating environment. You may choose to use the following examples as a guide to grant file permissions for the agent for IIS. This command and the syntax shown are subject to change by Microsoft at any time and without notice. For more information, go to the Microsoft Support (http://support.microsoft.com/) website, and search for "icacls"

**Follow these steps:**

1. Open a Command Prompt Window on your IIS web server.

**Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

2. Run the icacls command. Use the following example as a guide:

   ```
   icacls log_directory /grant IIS AppPool\application_pool_identity
   ```

   - **log_directory**
     Specifies the non default log directory to which you must grant permissions.

   - **application_pool_identity**
     Specifies the identity of the application pool associated with the application protected by CA Single Sign-On on your IIS web server.

3. Repeat Step 2 for each application pool identity on your IIS web server. For example, if you have two application pools, grant permissions to both.

4. If you have an IIS server farm using Shared Configuration, repeat Steps 1 through 3 for each IIS web server in the farm.
   The permissions are set.

## Change IIS Settings Manually for CA Single Sign-On Authentication Schemes Requiring Certificates

If you use CA Single Sign-On authentication schemes that request or require certificates, change the settings manually on your IIS web server for the following virtual directories:

- cert
- certoptional

**Follow these steps:**

1. Open IIS manager.
2. Expand your web server.
   The Application pools icon and Sites folder appear.
3. Expand Sites.
4. Expand the website associated with your authentication scheme that requires certificates.
   The siteminderagent virtual folder appears.
5. Expand the siteminderagent virtual folder.
6. Click the cert folder.
7. Double-click SSL Settings.
8. Select the Require SSL check box, and then click the Require option button.

9. Under Actions, click Apply
10. Click the certoptional folder.
11. Double-click SSL Settings.
12. Click the Accept option button.
13. Under Actions, click Apply.
14. Repeat Steps 3 through 14 for other websites on your IIS web server that require certificates.
15. For IIS server farms using Shared Configuration, repeat Steps 1 through 15 on each IIS web server in your farm.
    The settings are changed.

# Uninstall an IIS Agent

Before you remove the CA Single Sign-On Web Agent from a Windows operating environment, consider making backup copies of your registry settings and Web Agent configuration settings.

Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

**Note**: To remove the CA Single Sign-On Web Agent for IIS from a server farm, run the uninstall program on *each* node in the farm. Start by removing the Web Agent from the first web server in the farm, and then remove the Web Agent from all other nodes. The first server refers to the IIS web server where the shared configuration information is stored. A node refers to any IIS web servers which read the shared configuration from the first server.

**Follow these steps:**

1. Stop the web server.
2. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
3. Choose *one* of the following procedures:
   - To remove the Web Agent using the wizard, go to Step 4.
   - To remove the Web Agent using the console-based program, go to Step 9.

4. Click Start, Control Panel, Programs and Features.
5. Click CA CA Single Sign-On Web Agent *version*.
6. Click Uninstall/Change.
7. Review the information in the Uninstall CA Single Sign-On Web Agent dialog, then click Uninstall.
8. Wait for the wizard to finish, then go to Step 13.
9. Open a command-line window.
10. Navigate to the following directory.
    *web_agent_home*

    - ***web_agent_home***
      Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
      **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
      **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
      **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

11. Run the following command:

    ```
    ca-wa-uninstall.cmd -i console
    ```

12. Wait for the un–installation program to finish, then go to Step 13.
13. Start the web server.

**Important!** Delete the ZeroG registry file from the following location after uninstalling the Web Agent: C:\Program Files\ZeroG Registry\com.zerog.registry.xml

# Silently Remove an IIS Agent

The CA Single Sign-On Agent supports an unattended mode that uninstalls the agent. This option does not require any interaction from the end user.

1. Log in to your web server.
2. Open a Command Prompt window with Administrative privileges.

> **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

3. Run the configuration wizard silently to remove the configuration settings of the agents that you want to remove.

4. Run the following command:

```
web_agent_home\install_config_info\ca-wa-uninstall\uninstall.exe -
f installvariables.properties -i silent
```

> **Note**: If the path contains spaces, surround it with quotes.

- ***web_agent_home***
  Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
  **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
  **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
  **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32

The agent is removed from the web server.

5. For IIS server farms, repeat Steps 1 though 4 for each web server in your farm.

# Configure Multiple Agent Configurations for Application Pools

You can support independent agent configurations for application pools and their websites for Microsoft IIS Server. CA Single Sign-On provides a separate Web Agent configuration file for each application pool in your environment; this lets you configure different settings for different application pools and their websites. Configure an agent configuration object (ACO) for each application pool you want to manage independently.

## Configuration Using the GUI Mode

**Follow these steps**:

1. In the Default Agent Configuration Object step of the configuration wizard, select Manage Application Pool.

2. Click Next.

3. In the Configure Application Pool Step, perform the following steps for each application pool you want to configure:

    a. Select the websites.

    b. (Optional) Click Enable Agent to enable the agent in the WebAgent.conf file.

    c. Enter the ACO value that you defined for the application pool in Policy Server.

4. Click Next.

5. Review the web server configuration summary and click Install.

## Configuration Using the Unattended Mode

**Follow these steps**:

1. Configure the following properties of Microsoft IIS Server in the installer silent properties file:
   **AGENT_CONFIG_OBJ**
   Defines the name of the default ACO.
   **IS_IIS_SELECTED**
   Specifies that the Microsoft IIS Server is configured. Type true.
   **ENABLE_WEBAGENT_YES**
   Specifies whether the Agent is enabled for the default ACO. Enter 0 if the Agent is disabled. Enter 1 if the Agent is enabled.
   **MANAGE_APP_POOL**
   Specifies whether the application pools must be configured with default ACO or multiple ACOs. Enter 0 to configure application pools with default ACO. Enter 1 to configure application pools with multiple ACOs.
   **APPPOOL_CONFIGURE**
   **Valid only if MANAGE_APP_POOL is set to 1**
   Specifies which application pools must be managed independently. Define the ACO and Agent status of each application pool you want to manage independently in the following format:

   *Application_Pool_Name*,*ACO*,*Agent_Status*

   For example, consider that you have two application pools AppPool1, AppPool2 to be configured with the ACOs aco1, aco2, and that you want to enable the Agent for only AppPool1. Enter the following value for APPPOOL_CONFIGURE:

   ```
   AppPool1,aco1,true,AppPool2,aco2,false
   ```

   Use a comma "," to separate the values of multiple application pools. If you do not specify an application pool in APPPOOL_CONFIGURE, the default ACO is used for the application pool.
   **CONFIGURE_SITES**
   Specifies the web sites that must be configured. Separate multiple values by a comma ",". If you specify a web site that has been configured earlier, the configuration is overwritten.
   **UNCONFIGURE_SITES**
   Specifies the web sites that must be unconfigured. Separate multiple values by a comma ",".
   **Note**: If a web site has been configured earlier but you do not specify it in CONFIGURE_SITES AND UNCONFIGURE_SITES, the web site is preserved.

2. Run the configuration wizard:

   ```
   ca-wa-config.cmd –i silent –f silent_property_filename
   ```

After installation, CA Single Sign-On Agent creates a folder for each enabled application pool along with the LocalConfig.conf and WebAgent.conf files in the following location:

```
\webagent\win64\bin\IIS or \webagent\win32\bin\IIS
```

No folder is created for application pools using the default ACO.

During initialization, CA Single Sign-On reads the corresponding AppPool\WebAgent conf file path. If the file path does not exist, CA Single Sign-On reads the default WebAgent.conf file corresponding to the default ACO.

***Important!*** Configure separate Web Agent log and trace files for each ACO. Using the same WA log and trace files for multiple ACOs can cause synchronization issues during logging.

# Web Agent for Oracle iPlanet

The content in this section describes how to install and configure an agent on a Oracle iPlanet web server. Use the Table of Contents to access the content.

## Hardware Requirements for an Oracle iPlanet Agent

- **Windows operating environment requirements**
  agents operating on Windows operating environments require the following hardware:
  - CPU: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in the temporary location.
- **UNIX operating environment requirements**
  Agents operating on UNIX operating environments require the following hardware:
  - CPU:
    - Solaris operating environment: SPARC
    - Red Hat operating environment: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:

- 2-GB free disk space in the installation location.
- .5-GB free disk space in /temp.

> **Note**: Daily operation of the agent requires 10 MB of free disk space in /tmp. The agent creates files and named pipes under /tmp. The path to which these files and pipes are created cannot be changed.

# Policy Server Requirements for Oracle iPlanet Agents

Verify the following criteria:

- Your Policy Server is installed and configured.
- Your Policy server can communicate with the computer where you plan to install the agent.

To install and configure an agent, a Policy Server requires at least the following items:

- A CA Single Sign-On administrator that has the right to register trusted hosts.
  A trusted host is a client computer where one or more CA Single Sign-On Agents are installed and registered with the Policy Server. The CA Single Sign-On administrator must have permissions to register trusted hosts with the Policy Server. Registering a trusted host creates a unique trusted host name object on the Policy Server.
- An Agent identity
  An Agent identity establishes a mapping between the Policy Server and the name or IP address of the web server instance hosting an Agent. You define an Agent identity from the Agents object in the Administrative UI. You assign it a name and specify the Agent type as a Web Agent.
- A Host Configuration Object (HCO)
  The host configuration object on the Policy Server defines the communication between the agent and the Policy Server that occurs after an initial connection. The Initial connections use the parameters in the SmHost.conf file.
- Agent Configuration Object (ACO)
  This object includes the parameters that define the agent configuration. All CA Single Sign-On agents require at least one of the following configuration parameters that are defined in the ACO:
  - **AgentName**
    Defines the identity of the web agent. This identity links the name and the IP address or FQDN of each web server instance hosting an Agent.
    The value of the DefaultAgentName is used instead of the AgentName parameter if any of the following events occur:
    - The AgentName parameter is disabled.
    - The value of AgentName parameter is empty.
    - The values of the AgentName parameter do *not* match any existing agent object.

Note: This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add each value to a separate line in the file.
**Default:** No default
**Limit**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
**Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.
**Example:** myagent1,192.168.0.0 (IPV4)
**Example:** myagent2, 2001:DB8::/32 (IPV6)
**Example:** myagent,www.example.com
**Example** (multiple AgentName parameters): AgentName1, AgentName2, AgentName3. The value of each AgentName*number* parameter is limited to 4,000 characters.

- **DefaultAgentName**
  Defines a name that the agent uses to process requests. The value for DefaultAgentName is used for requests on an IP address or interface when no agent name value exists in the AgentName parameter.
  If you are using virtual servers, you can set up your CA Single Sign-On environment quickly by using a DefaultAgentName. Using DefaultAgentName means that you do not need to define a separate agent for each virtual server.

  **Important!** If you do not specify a value for the DefaultAgentName parameter, then the value of the AgentName parameter requires every agent identity in its list. Otherwise, the Policy Server cannot tie policies to the agent.

  **Default:** No default.
  **Limit**: Multiple values are allowed.
  **Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.

# How to Prepare for a Web Agent Installation on an Oracle iPlanet Web Server

To prepare for a CA Single Sign-On agent installation on an Oracle iPlanet server, use the following process:

1. Verify that you have an account with one of the following types of privileges for your web server:
   - Administrative privileges (for the Windows operating environment)
   - Root privileges (for the UNIX or Linux operating environments)

2. Configure the appropriate additional settings that a CA Single Sign-On Agent requires using *one* of the following lists:

   - Oracle iPlanet web server preparations for UNIX operating environments (see page 612).
   - Oracle iPlanet web server preparations for Linux operating environments (see page 613).

# Oracle iPlanet Web Server Preparations for UNIX

**Contents**



## Set the DISPLAY For CA Single Sign-On Agent Installations on UNIX

If you are installing the CA Single Sign-On Agent on a UNIX system from a remote terminal, such as a Telnet or Exceed terminal, be sure the DISPLAY variable is set for the local system. For example, if your machine is 111.11.1.12, set the variable as follows:

DISPLAY=111.11.1.12:0.0

export DISPLAY

**Note:** You can also install the agent using the console mode installation, which does not require the X window display mode.

## Required Solaris Patches

Before installing a CA Single Sign-On Agent on a Solaris computer, install the following patches:

- **Solaris 9**
  Requires patch 111711-16.
- **Solaris 10**
  Requires patch 119963-08.

You can verify installed patch versions by logging in as the root user and executing the following command:

```
showrev -p | grep patch_id
```

To locate Solaris patches, go to the Oracle Solution Center.

## AIX Requirements

CA Single Sign-On agents running on AIX systems require the following configurations:

- To run a rearchitected (framework) CA Single Sign-On agent for Oracle iPlanet on an AIX system, your C/C++ runtime environment must be version 8.0.0.0.

# Oracle iPlanet Web Server Preparations for Linux

This content describes software packages and libraries that are required on Linux systems before installing a Web Agent for Oracle iPlanet.

## Required Linux Software Packages

The following software packages are required to install Web Agents on 64-bit Linux systems

- Binutils 2.17
- GCC 4.7.2

## Required Linux Libraries

CA Single Sign-On requires certain Linux libraries for components that operate on Linux. We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

The following list describes the commands to install the required libraries on the host system:

**Red Hat 5.x**

```
yum install -y compat-gcc-34-c++
yum install -y libidn.i686
yum install -y libstdc++.i686
yum install -y ncurses-libs.i686
```

**Red Hat 6.x**

```
yum install -y libstdc++.i686
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y ncurses-libs.i686
yum install -y libXrender.i686
yum install -y libXtst.i686
```

**Additional Packages for Red Hat 6.x 64-bit**

```
yum install -y libXau.i686
yum install -y libXext.i686
yum install -y libxcb.i686
yum install -y compat-libstdc++-33.i686
yum install -y compat-db42.i686
yum install -y compat-db.i686
yum install -y compat-db43.i686
yum install -y libXi.i686
yum install -y libX11.i686
yum install -y libXtst.i686
```

```
yum install -y libXrender.i686
yum install -y libXft.i686
yum install -y libXt.i686
yum install -y libXp.i686
yum install -y libstdc++.i686
yum install -y libICE.i686
yum install -y compat-libtermcap.i686
yum install -y libidn.i686
yum install -y libSM.i686
yum install -y libuuid.i686
```

If the correct library is unavailable, CA Single Sign-On displays the following error:

```
java.lang.UnsatisfiedLinkError
```

# Install and Configure an Oracle iPlanet Agent on Windows

This section contains the following topics:

- How to Install and Configure the Agent for Oracle iPlanet on Windows (see page 614)
- Run a Silent Installation and Configuration for iPlanet Agents on Windows (see page 621)

## How to Install and Configure the Agent for Oracle iPlanet on Windows

**Contents**



### Gather the Information for the Installation Program

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path

### Run the Installation Program on Windows

The installation program for the agent installs the agent on one computer at a time using the Windows operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation programs also create a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy the Web Agent installation executable file to a temporary directory on your web server.
2. Do *one* of the following steps:
   - For wizard-based installations, right-click the installation executable file, and then select Run as Administrator.
   - For console-based installations, open a command line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```
3. Use the information that you gathered previously to complete the installation.

## Gather the Information Required by the Configuration Program on Windows

Gather the following information about the environment for the product before running the configuration program for the agent:

- **Register Host**
  Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.
  **Default**:Yes
  **Options**: Yes, No
- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This CA Single Sign-On user account requires privileges to register trusted hosts.
- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.
- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.
- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.

- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.

- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.
  **Example**: (IPV4) 192.168.1.105
  **Example**: (IPV4 with the port number) 192.168.1.105:44443
  **Example**: (IPV6) 2001:DB8::/32
  **Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.
    **Default**: FIPS Compatibility/AES Compatibility
    FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

    **Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the CA Single Sign-On agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.

  - **Default: SmHost.conf**

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Run the Web Agent Configuration Program on Windows

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console modes once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:
   *web_agent_home*\install_config_info

   - ***web_agent_home***
     Indicates the directory where the CA Single Sign-On agent is installed on your web server.
     **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
     **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
     **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, go to Step 3.

   - For a console-based configuration, go to Step 5.

3. Right-click the following executable, and then select Run as Administrator:

   ```
   ca-wa-config.exe
   ```

4. Go to Step 8.

5. Open a Command Prompt window with Administrator privileges.

6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```

7. Go to Step 8.

8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## Apply Changes to Oracle iPlanet Configuration Files for SunOne 6.1 Servers

The Agent Configuration Wizard modifies the default obj.conf, and mime.types files that the Oracle iPlanet web server uses.

If you are using version 6.1 of a SunOne web server, and you plan to use the Oracle iPlanet Administration console, apply the changes to these files *before* using the console. If you do not apply the changes using the console first, the changes that are made for your CA Single Sign-On configuration could be corrupted. If you lose your configuration, run the configuration program again.

The agent adds settings to the obj.conf file of the Oracle iPlanet web server when the Agent is configured to support an advanced authentication scheme. CA Single Sign-On does *not* remove these settings later. Edit the obj.conf file manually to remove any obsolete settings.

**Follow these steps:**

1. Log in to the Oracle iPlanet Administration Server console.

2. From the Servers tab, select the web server with the CA Single Sign-On agent installed and click Manage.

3. In the right corner of the dialog, click Apply.
   A warning message about loading the modified configuration files appears.

4. Click Load Configuration Files.

5. Exit the console.

6. Restart the web server.

7. Optimize the Agent for Oracle iPlanet by tuning the shared memory segments.
   The CA Single Sign-On changes are applied.

## Configure Non-Default Server Instances, Virtual Servers, or Reverse Proxies for Oracle iPlanet Web Servers

The Web Agent Configuration wizard only configures the default instance of your Oracle iPlanet web server. To configure a different instance of the Oracle iPlanet web server, manually edit the obj.conf file that is associated with that server instance. Examples of server instances that need manual configuration include:

- Servers installed in a non-default directory
- Servers that you want to configure as a reverse proxy. We recommend configuring the reverse proxy using your Oracle iPlanet interface *before* adding the CA Single Sign-On settings to the obj.conf file.
  The Agent Configuration wizard *only* modifies the *default* obj.conf file on the Oracle iPlanet (formerly Sun Java System) web server. To protect other instances or reverse proxy deployments with CA Single Sign-On, copy the settings from the default obj.conf file to any respective *instance_name*-obj.conf files. For example, your web server created an obj.conf file when you installed it, but you later added a server instance named my_server.example.com ( http://my_server.example.com/). To protect resources on my_server.example.com (http://my_server.example.com/) , copy the settings the wizard added from the obj.conf file to the my_server.example.com (http://my_server.example.com/)-obj.conf file.
- Virtual servers on the same computer

**Note**: SunOne/Sun Java 7.0 web servers do *not* require these manual configuration steps.

**Follow these steps:**

1. Locate the directory of the server instance you want to configure.
2. Open the obj.conf file with a text editor.
3. Locate the following line:

   ```
   <Object name="default">
   ```

4. Insert a new line below the previous one, and then add the following text:

   ```
   AuthTrans fn="SiteMinderAgent"
   ```

5. Locate the following line:

   ```
   AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
   ```

6. Insert a new line below the previous one, and then add the following text:

   ```
   NameTrans fn="pfx2dir" from="/siteminderagent/pwcgi" dir="web_agent_home
   /pw" name="cgi"
   NameTrans fn="pfx2dir" from="/siteminderagent/pw" dir="web_agent_home/pw"
   NameTrans fn="pfx2dir" from="/siteminderagent/jpw" dir="web_agent_home/jpw"
   NameTrans fn="pfx2dir" from="/siteminderagent/redirectjsp" dir="web_agent_home
   /affwebservices/redirectjsp"
   ```

```
NameTrans fn="pfx2dir" from="/siteminderagent/certoptional" dir="web_agent_home
/samples"
NameTrans fn="pfx2dir" from="/siteminderagent" dir="web_agent_home/samples"
NameTrans fn="pfx2dir" from="/siteminderagent/pwservlet" dir=web_agent_home/jpw"
```

7. **web_agent_home**
   Indicates the directory where the CA Single Sign-On agent is installed on your web server.
   **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
   **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
   **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32

8. Locate the following line:

   ```
   NameTrans fn="ntrans-j2ee" name="j2ee"
   ```

9. Insert a new line below the previous one, and then add the following text:

   ```
   PathCheck fn="SmRequireAuth"
   ```

10. Remove the following line:

    ```
    NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Program Files/Sun/WebServer7.0
    /lib/icons" name="es-internal"
    ```

11. Locate the following line:

    ```
    ObjectType fn="force-type" type="text/plain"
    ```

12. Insert a new line below the previous one, and then add the following text:

    ```
    Service method="(GET|POST)" fn="SmAdvancedAuth"
    ```

13. Save the obj.conf file.
    The Oracle iPlanet web server is manually configured.

## (Optional) Improve Server Performance with httpd.conf File Changes

You can improve server performance by modifying the default configuration settings in the httpd.conf file; however, these changes are *not* required:

**Follow these steps:**

1. For Oracle iPlanet web servers, assign a higher priority level to your Apache20WebAgent.dll file than any other auth modules or access modules on your web server.

2. For low-traffic websites, define the following directives:

   - Set MaxRequestsPerChild>1000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >5
   - MaxSpareServers>10
   - StartServers=MinSpareServers>5

3. For high-traffic websites, define the following directives:

- Set MaxRequestsPerChild>3000 *or* Set MaxRequestsPerChild=0
- MinSpareServers >10
- MaxSpareServers>15
- StartServers=MinSpareServers>10

## Run a Silent Installation and Configuration for iPlanet Agents on Windows

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The CA Single Sign-On Web Agent Installation wizard.
   b. The CA Single Sign-On Web Agent Configuration wizard.
2. Locate the following file on your first web server:

   ```
   web_agent_home\install_config_info\ca-wa-installer.properties
   ```

   **Note**: If the path contains spaces, surround it with quotes.

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On agent is installed on your web server.
   **Default** (Windows 32-bit installations only): C:\Program Files\CA\webagent
   **Default** (Windows 64-bit installations only): C:\Program Files\CA\webagent\win64
   **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64]): C:\Program Files (x86)\webagent\win32
4. Perform each of the following steps on the other web servers in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.

b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:

   - The CA Single Sign-On Web Agent Installation executable file.
   - CA Single Sign-On ca-wa-installer properties file.

c. Open a Command Prompt window with Administrative privileges in the temporary directory.

d. Run the following command:

   ```
   agent_executable -f properties_file -i silent
   ```

   The CA Single Sign-On agent is installed and configured on the subsequent server silently.

e. (Optional) Delete the temporary directory from your subsequent web server.

5. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Install and Configure an Oracle iPlanet Agent on UNIX/Linux

This section contains the following topics:



## How to Install and Configure an Agent for Oracle iPlanet on UNIX Linux

**Contents**



Installing and configuring the CA Single Sign-On Agent for Oracle iPlanet involves several separate procedures. To install and configure the Agent for Oracle iPlanet, use the following process:

1. Gather the information for the installation program (see page 623).
2. Run the wizard based installation program (see page 623).
3. Gather the information for the configuration program (see page 624).
4. Source the agent environment script (see page 626).
5. Set the library path variable (see page 627).
6. Run the wizard based configuration program (see page 627).
7. (Optional) Install and configure additional Agents for Oracle IPlanet silently.
8. Determine if your Agent for Oracle iPlanet requires any of the following additional configuration steps:
   - (For SunOne 6.1 web servers only) If you want to use the Oracle iPlanet Administration Server console, apply the changes to the configuration files of the Oracle iPlanet web server (see page ).
   - (Except SunOne 7.0/Sun Java 7.0 web servers) Manually configure any nondefault server instances, reverse proxies, or virtual servers for (see page ).
   - For Solaris 9 SP3 and Solaris 10, modify the startup script (see page ).

## Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Run the Installation Program on UNIX Linux

The installation program for the CA Single Sign-On agent installs the agent on one computer at a time using the UNIX or Linux operating environments. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Copy CA Single Sign-On agent installation executable file to a temporary directory on your web server.
2. Log in as a root user.

3. Do *one* of the following steps:

   - For wizard-based installations, run the installation executable file.
   - For console-based installations, open a command-line window and run the executable as shown in the following example:

     ```
     executable_file_name.exe -i console
     ```

4. Use the information from your agent Installation worksheet to complete the installation program.

## Gather the Information that the Configuration Program Requires on UNIX Linux

Gather the following information about the environment for the product before running the configuration program for the agent:

Register Host

Indicates whether you want to register a trusted host. This registration creates a trusted host object in the Policy Server and an SmHost.conf file on the web server. The agent uses this information to make an initial connection to Policy Servers when it starts. Register each agent instance as a trusted host only once.

**Default**:Yes

**Options**: Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user with Administrative privileges that is already defined in the Policy Server. This CA Single Sign-On user account requires privileges to register trusted hosts.
- **Admin Password**
  Specifies a password for the Admin User Name that is already defined in the Policy Server.
- **Confirm Admin Password**
  Repeats the password entered in the Admin Password field. This value verifies the password for the Admin User Name already defined in the Policy Server.
- **Trusted Host Object Name**
  Specifies a unique name for the trusted host you are registering. This trusted host object is stored on the Policy Server.
- **Host Configuration Object**
  Specifies the name of a Host Configuration Object that is already defined in the Policy Server. After the agent initially connects to a Policy Server (using the SmHost.conf file settings), subsequent connections use the settings from the Host Configuration Object.
- **Policy Server IP Address**
  Specifies the Internet Protocol address of the Policy Servers to which the agent attempts to connect upon startup. If your Policy Server is behind a firewall, specify a port number also.
  If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.

**Example**: (IPV4) 192.168.1.105
**Example**: (IPV4 with the port number) 192.168.1.105:44443
**Example**: (IPV6) 2001:DB8::/32
**Example**: (IPV6) [2001:DB8::/32]:44443

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:

  - **FIPS Compatibility/AES Compatibility**
    Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does *not* require the use of FIPS-compliant algorithms, use this option.

  - **FIPS Migration/AES Migration**
    Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.

  - **FIPS Only/AES Only**
    Uses *only* FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does *not* interoperate with, *nor* is backwards-compatible with, previous versions of CA Single Sign-On.

  - **Default: FIPS Compatibility/AES Compatibility**

  FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

  **Important!** Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the CA Single Sign-On agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored.
  **Default**: *web_agent_home*\config

- **Enable Shared Secret Rollover**
  Select this check box to change the shared secret that the Policy Server uses to encrypt communications to the Web Agents.

- **Select Servers**
  Indicates the web server instances that the configuration program finds on the computer. Select the check boxes of the instances you want to configure. Clear the check boxes of those instances from which you want to remove CA Single Sign-On protection.

- **Agent Configuration Object Name**
  Specifies the name of an agent configuration object (ACO) already defined on the Policy Server.
  **Default**: AgentObj

- **Advanced Authentication Scheme Dialog**
  Specifies the advanced authentication scheme for the web server instances you selected previously.

## Source the Agent Environment Script on UNIX or Linux

The agent installation program creates an environment script, **ca_wa_env.sh** in the following directory:

*web_agent_home*/ca_wa_env.sh

*web_agent_home* indicates the directory where the Agent is installed. The default UNIX/LINUX location for the script is:

opt/ca/webagent

For RHEL 7, include the content of the source script in the directory:

/etc/sysconfig/httpd

The following is a sample of the modified script in the directory /etc/sysconfig/httpd. Strings in **bold** are in effect and others are commented out.

Note the following:

- Replace any ${*VARIABLE*} with the actual value.
- To determine the values for the variables ${LD_LIBRARY_PATH} and ${PATH}, use the **env** command before you add the script contents.

```
NETE_WA_ROOT=/opt/CA/webagent
export NETE_WA_ROOT
NETE_WA_PATH=/opt/CA/webagent/bin
#NETE_WA_PATH=${NETE_WA_ROOT}/bin
export NETE_WA_PATH
 CAPKIHOME=/opt/CA/webagent/CAPKIexport CAPKIHOME
LD_LIBRARY_PATH=/opt/CA/webagent/bin:/opt/CA/webagent/bin/thirdparty#LD_LIBRARY_PATH=$
{NETE_WA_ROOT}/bin:${NETE_WA_ROOT}/bin/thirdparty:${LD_LIBRARY_PATH}
export LD_LIBRARY_PATH
PATH=/opt/CA/webagent/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr
/bin:/root/bin
#PATH=/opt/CA/webagent/bin:${PATH}
#PATH=${NETE_WA_PATH}:${PATH}
export PATH
```

For most Apache-based web servers, source this script *before* doing any of the following tasks:

- Running the agent configuration program.
- Starting the web server.

**Note**: If you perform *all* the previous tasks in the *same* shell, only source the script *once*.

For the embedded Apache web server included with RedHat Linux, do *one* of the following tasks:

- Source the script *before* starting the httpd service.
- Source the script in the following file instead of starting it manually each time:

  `/etc/init.d/htppd`

## Set the Library Path Variable on UNIX or Linux

Set the library path variable on UNIX or Linux systems before running the agent configuration program.

The following table lists the library path variables for the various UNIX and Linux operating environments:

| Operating System | Name of Library Path Variable |
|---|---|
| AIX | LIBPATH |
| Linux | LD_LIBRARY_PATH |
| Solaris | LD_LIBRARY_PATH |

Set the value of the library path variable to the *agent_home*/bin directory.

- ***agent_home***
  Indicates the directory where the Agent is installed.

## Run the Web Agent Configuration Program on UNIX Linux

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:
   *web_agent_home/*install_config_info

   - **web_agent_home**
     Indicates the directory where the CA Single Sign-On Agent is installed.
     **Default** (UNIX/Linux installations): /opt/ca/webagent

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, go to Step 3.

- For a console-based configuration, go to Step 5.

3. Run the following executable file:

   ```
   ca-wa-config.bin
   ```

4. Go to Step 8.

5. Open a Command Prompt window with root privileges.

6. Navigate to the executable file listed previously, and then run it with the following switch:

   ```
   -i console
   ```

7. Go to Step 8.

8. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## Apply CA Single Sign-On Changes to Oracle iPlanet Configuration Files with Oracle iPlanet Administration Server Console for SunOne 6.1 Servers

The Agent Configuration Wizard modifies the default obj.conf, and mime.types files that the Oracle iPlanet web server uses.

If you are using version 6.1 of a SunOne web server, and you plan to use the Oracle iPlanet Administration console, apply the changes to these files *before* using the console. If you do not apply the changes using the console first, the changes that are made for your CA Single Sign-On configuration could be corrupted. If you lose your configuration, run the configuration program again.

**Note:** The agent adds settings to the obj.conf file of the Oracle iPlanet web server when the Agent is configured to support an advanced authentication scheme. CA Single Sign-On does *not* remove these settings later. Edit the obj.conf file manually to remove any obsolete settings.

**Follow these steps:**

1. Log in to the Oracle iPlanet Administration Server console.

2. From the Servers tab, select the web server with the CA Single Sign-On agent installed and click Manage.

3. In the right corner of the dialog, click Apply.
   A warning message about loading the modified configuration files appears.

4. Click Load Configuration Files.

5. Exit the console.

6. Restart the web server.

7. Optimize the Agent for Oracle iPlanet by tuning the shared memory segments.
   The CA Single Sign-On changes are applied.

## Manually Configure Non-default Server Instances, Virtual Servers, or Reverse Proxies for Oracle iPlanet Web Servers

The CA Single Sign-On Web Agent Configuration wizard only configures the default instance of your Oracle iPlanet web server. To configure a different instance of the Oracle iPlanet web server for CA Single Sign-On, manually edit the obj.conf file that is associated with that server instance. Examples of server instances that need manual configuration include:

- Servers installed in a nondefault directory
- Servers that you want to configure as a reverse proxy. We recommend configuring the reverse proxy using your Oracle iPlanet interface *before* adding the CA Single Sign-On settings to the obj.conf file.

  Note: The CA Single Sign-On Agent Configuration wizard only modifies the default obj.conf file on the Oracle iPlanet (formerly Sun Java System) web server. To protect other instances or reverse proxy deployments with CA Single Sign-On, copy the CA Single Sign-On settings from the default obj.conf file to any respective instance_name-obj.conf files. For example, your web server created an obj.conf file when you installed it, but you later added a server instance named my_server.example.com. To protect resources on my_server.example.com with CA Single Sign-On, copy the CA Single Sign-On settings the wizard added from the obj.conf file to the my_server.example.com-obj.conf file.

- Virtual servers on the same computer

**Note**: SunOne 7.0 web servers do *not* require these manual configuration steps.

**Follow these steps:**

1. Locate the directory of the server instance you want to configure.
2. Open the obj.conf file with a text editor.
3. Locate the following line:

   ```
   <Object name="default">
   ```

4. Insert a new line below the previous one, and then add the following text:

   ```
   AuthTrans fn="SiteMinderAgent"
   ```

5. Locate the following line:

```
AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
```

6. Insert a new line below the previous one, and then add the following text:

```
NameTrans fn="pfx2dir" from="/siteminderagent/pwcgi" dir="web_agent_home
/pw" name="cgi"
NameTrans fn="pfx2dir" from="/siteminderagent/pw" dir="web_agent_home/pw"
NameTrans fn="pfx2dir" from="/siteminderagent/jpw" dir="web_agent_home/jpw"
NameTrans fn="pfx2dir" from="/siteminderagent/redirectjsp" dir="web_agent_home
/affwebservices/redirectjsp"
NameTrans fn="pfx2dir" from="/siteminderagent/certoptional" dir="web_agent_home
/samples"
NameTrans fn="pfx2dir" from="/siteminderagent" dir="web_agent_home/samples"
```

7. Locate the following line:

```
NameTrans fn="ntrans-j2ee" name="j2ee"
```

8. Insert a new line below the previous one, and then add the following text:

```
PathCheck fn="SmRequireAuth"
```

9. Remove the following line:

```
NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Program Files/Sun/WebServer7.0
/lib/icons" name="es-internal"
```

10. Locate the following line:

```
ObjectType fn="force-type" type="text/plain"
```

11. Insert a new line below the previous one, and then add the following text:

```
Service method="(GET|POST)" fn="SmAdvancedAuth"
```

12. Save the obj.conf file.
    The Oracle iPlanet web server is manually configured.

## Modify the Oracle iPlanet Startup Script to Prevent Crashes when the Server Stops

The Oracle iPlanet server can sometimes crash when shutting down in the following operating environments:

- Solaris 9 SP3
- Solaris 10

Modify the startserv script to prevent the Oracle iPlanet web server from crashing when shuttng down.

**Follow these steps:**

1. Open the following file with a text editor:

```
sunone_instance_directory/bin/startserv
```

   - ***sunone_instance_directory***
     Indicates the directory of the SunOne web server instance.

2. Locate the following line:

```
LIBUMEM_32=/usr/lib/libumem.so
```

3. Add a comment character in the beginning of the previous line. See the following example:

   ```
   #LIBUMEM_32=/usr/lib/libumem.so
   ```

4. Locate the following line:

   ```
   LIBUMEM_64=/usr/lib/64/libumem.so
   ```

5. Add a comment character in the beginning of the previous line. See the following example:

   ```
   #LIBUMEM_64=/usr/lib/64/libumem.so
   ```

6. Save the file and close the text editor.
   The Oracle iPlanet startup script is modified.

### Manage Content Compression

If you enabled compression at server level and want to compress the data generated by Web Agent, set the **enablecompression** ACO parameter to **Yes**.

## Run a Silent Installation and Configuration for Oracle iPlanet Agents on UNIX Linux

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):

   a. The CA Single Sign-On Web Agent Installation wizard.

   b. The CA Single Sign-On Web Agent Configuration wizard.

2. Locate the following file on your first web server:

   ```
   web_agent_home/install_config_info/ca-wa-installer.properties
   ```

   **Note**: If the path contains spaces, surround it with quotes.

3. ***web_agent_home***
   Indicates the directory where the CA Single Sign-On agent is installed on your web server.

4. Perform each of the following steps on the other web servers in your environment:

**Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

a. Create a temporary directory on the subsequent web server.

b. Copy the following files from the first web server (Steps 1 and 2) to the temporary directory on your subsequent web server:

- The CA Single Sign-On Web Agent Installation executable file.
- CA Single Sign-On ca-wa-installer properties file.

c. Open a Command Prompt window with Administrative privileges in the temporary directory.

d. Run the following command:

```
agent_executable -f properties_file -i silent
```

The CA Single Sign-On agent is installed and configured on the subsequent server silently.

e. (Optional) Delete the temporary directory from your subsequent web server.

5. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wa-installer.properties file specify.

# Uninstall an Oracle iPlanet Agent from a Windows Operating Environment

Before you un–install the CA Single Sign-On Web Agent from a Windows operating environment, consider making backup copies of your registry settings and Web Agent configuration settings.

Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

**Follow these steps:**

1. Stop the web server.
2. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.

3. Choose *one* of the following procedures:
   - To remove the Web Agent using the wizard, go to Step 4.
   - To remove the Web Agent using the console-based program, go to Step 9.
4. Click Start, Control Panel, Programs and Features.
   A list of installed programs appears.
5. Click CA Single Sign-On Web Agent *version_number*.
6. Click Uninstall/Change.
   The uninstallation wizard appears.
7. Review the information in the Uninstall CA Single Sign-On Web Agent dialog, then click Uninstall.
   The wizard removes the web agent.
8. Wait for the wizard to finish, then go to Step 13.
9. Open a command-line window.
10. Navigate to the following directory.

    ```
    web_agent_home
    ```

    - ***web_agent_home***
      Indicates the directory where the CA Single Sign-On Agent is installed on your web server.
      **Default** (Windows 32-bit installations of CA Single Sign-On IIS Web Agents only): C:\Program Files\CA\webagent
      **Default** (Windows 64-bit installations [CA Single Sign-On Web Agents for IIS only]): C:\Program Files\CA\webagent\win64
      **Default** (Windows 32-bit applications operating on 64-bit systems [Wow64 with CA Single Sign-On Web Agents for IIS only]): C:\Program Files (x86)\webagent\win32
11. Run the following command:

    ```
    ca-wa-uninstall.cmd -i console
    ```

12. Wait for the un–installation program to finish, then go to Step 13.
13. Start the web server.

**Important!** Delete the ZeroG registry file from the following location after uninstalling the Web Agent: C:\Program Files\ZeroG Registry\com.zerog.registry.xml

# Uninstall an Oracle iPlanet Agent from a UNIX System

These instructions are for GUI and Console Mode removal.

**Note:** Removing a Web Agent from a 64-bit SuSE Linux 10 system requires additional preparations.

Be aware of the following:

- All Web Agents for all installed web servers will be uninstalled.
- The Password Services and Forms directories, (pw_default, jpw_default, samples_default) will be removed. However, the non-default copies of these directories (pw, jpw, samples) are not removed because these directories may contain customized files.

The steps for the two modes are the same, with these exceptions for Console Mode:

- Select an option by entering a corresponding number.
- Press Enter after each step to proceed through the process.

**Note:** Before you uninstall, we recommend copying your agent configuration settings to have as a backup.

**Follow these steps:**

1. Stop the web server.
2. Log in to the UNIX system.
3. Run the configuration wizard to remove the configuration settings of the agents that you want to remove.
4. Navigate to the directory where the Web Agent is installed: *web_agent_home* /install_config_info/ca-wa-uninstall
5. Verify that the unistallation program has execute permissions. For example, use the following command:

   ```
   chmod +x ca-wa-uninstall
   ```

6. From a console window, enter one of the following commands:

   - GUI mode: ./ca–wa–uninstall
   - Console mode: ./ca–wa–uninstall -i console

   The uninstallation program starts.

7. Read the information in the dialog to confirm the removal of the Web Agent, then click Uninstall. The Web Agent is removed from the system.
8. Click Done to exit the uninstallation program.
9. Change to your home directory (the current directory has been deleted).

10. Restart the web servers.

> **Note**: For Oracle iPlanet web servers, the obj.conf, magnus.conf, and mime.types files are restored to their original settings that existed before the agent was installed.

# Web Services Security Agent for Apache-based Servers

The following sections detail how to install and configure a WSS agent on an Apache-based web server.

## Policy Server Requirements for WSS Agents for Apache

Verify that:

- Your Policy Server is installed and configured.
- Your Policy server can communicate with the computer where you plan to install the agent.

To install and configure a CA Single Sign-On agent, a Policy Server requires at least the following items:

- A CA Single Sign-On administrator that has the right to register trusted hosts.
  A trusted host is a client computer where one or more CA Single Sign-On Agents are installed and registered with the Policy Server. Registering a trusted host creates a unique trusted host name object on the Policy Server.
- An Agent identity
  An Agent identity establishes a mapping between the Policy Server and the name or IP address of the web server instance hosting an Agent. You define an Agent identity from the Agents object in the Administrative UI. You assign it a name and specify the Agent type as a Web Agent.
- A Host Configuration Object (HCO)
  The host configuration object on the Policy Server defines the communication between the agent and the Policy Server that occurs after an initial connection. The Initial connections use the parameters in the SmHost.conf file.
- Agent Configuration Object (ACO)
  This object defines the agent configuration. All CA Single Sign-On agents require at least one of the following configuration parameters:

- **AgentName**
  This identity links the name and the IP address or FQDN of each web server instance hosting an Agent.
  The value of the DefaultAgentName is used instead of the AgentName parameter if any of the following events occur:

  - The AgentName parameter is disabled.
  - The value of AgentName parameter is empty.
  - The values of the AgentName parameter do *not* match any existing agent object.

  This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add each value to a separate line in the file.
  **Default:** No default
  **Limit**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
  **Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.
  **Example:** myagent1,192.168.0.0 (IPV4)
  **Example:** myagent2, 2001:DB8::/32 (IPV6)
  **Example:** myagent,www.example.com
  **Example** (multiple AgentName parameters): AgentName1, AgentName2, AgentName3. The value of each AgentName*number* parameter is limited to 4,000 characters.

- **DefaultAgentName**
  Defines a name that the agent uses to process requests. The value for DefaultAgentName is used for requests on an IP address or interface when no agent name value exists in the AgentName parameter.
  If you are using virtual servers, you can set up your CA Single Sign-On environment quickly by using a DefaultAgentName. Using DefaultAgentName means that you do not need to define a separate agent for each virtual server.
  **Important!** If you do not specify a value for the DefaultAgentName parameter, then the value of the AgentName parameter requires every agent identity in its list. Otherwise, the Policy Server cannot tie policies to the agent.
  **Default:** No default.
  **Limit**: Multiple values are allowed.
  **Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.

# Hardware Requirements for CA SiteMinder® Agents

Computers hosting CA Single Sign-On agents require the following hardware:

- **Windows operating environment requirements**
  agents operating on Windows operating environments require the following hardware:

- CPU: x86 or x64
- Memory: 2-GB system RAM.
- Available disk space:
  - 2-GB free disk space in the installation location.
  - .5-GB free disk space in the temporary location.

- **UNIX operating environment requirements**
  Agents operating on UNIX operating environments require the following hardware:
  - CPU:
    - Solaris operating environment: SPARC
    - Red Hat operating environment: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in /temp.

**Note**: Daily operation of the agent requires 10 MB of free disk space in /tmp. The agent creates files and named pipes under /tmp. The path to which these files and pipes are created cannot be changed.

# Apache-based server Preparations for Windows operating environments

**Contents**



## Install an Apache Web Server on Windows as a Service for All Users

When an Apache-based web server is installed using a single user account, the Agent configuration cannot detect the Apache-based web server installation.

To correct this problem, select the following option when you install an Apache-based web server on a Windows operating environment:

`"install as a service, available for all users".`

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based Web Agents

For Apache-based web server agents (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

# Apache-based Server Preparations for WSS Agents on UNIX

**Contents**



## Set the DISPLAY For CA Single Sign-On Agent Installations on UNIX

If you are installing the CA Single Sign-On Agent on a UNIX system from a remote terminal, such as a Telnet or Exceed terminal, be sure the DISPLAY variable is set for the local system. For example, if your machine is 111.11.1.12, set the variable as follows:

DISPLAY=111.11.1.12:0.0

export DISPLAY

**Note:** You can also install the agent using the console mode installation, which does not require the X window display mode.

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based CA Single Sign-On Agents

For any agents for Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

## Required Solaris Patches

Before installing a CA Single Sign-On Agent on a Solaris computer, install the following patches:

- **Solaris 9**
  Requires patch 111711-16.
- **Solaris 10**
  Requires patch 119963-08.

You can verify installed patch versions by logging in as the root user and executing the following command:

```
showrev -p | grep patch_id
```

To locate Solaris patches, go to the Oracle Solution Center.

## AIX Requirements

CA Single Sign-On agents running on AIX systems require the following components:

- To run a rearchitected (framework) CA Single Sign-On Apache-based agent on an AIX system, your C/C++ runtime environment must be version 8.0.0.0.

# Apache-based Server Preparations for WSS Agents on Linux

This content describes steps required before installing a WSS Agent on Linux.



## Verify Required Linux Software Packages

The following software packages are required to install Web Agents on 64-bit Linux systems:

- Binutils 2.17
- GCC 4.1.0

## Verify Required Linux Libraries

CA Single Sign-On requires certain Linux libraries for components that operate on Linux. We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

The following list describes the commands to install the required libraries on the host system:

**Red Hat 5.x**

```
yum install -y compat-gcc-34-c++
yum install -y libidn.i686
yum install -y libstdc++.i686
yum install -y ncurses-libs.i686
```

**Red Hat 6.x**

```
yum install -y libstdc++.i686
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y ncurses-libs.i686
yum install -y libXrender.i686
yum install -y libXtst.i686
```

**Additional Packages for Red Hat 6.x 64-bit**

```
yum install -y libXau.i686
yum install -y libXext.i686
yum install -y libxcb.i686
yum install -y compat-libstdc++-33.i686
yum install -y compat-db42.i686
yum install -y compat-db.i686
yum install -y compat-db43.i686
yum install -y libXi.i686
yum install -y libX11.i686
yum install -y libXtst.i686
yum install -y libXrender.i686
yum install -y libXft.i686
yum install -y libXt.i686
yum install -y libXp.i686
yum install -y libstdc++.i686
yum install -y libICE.i686
yum install -y compat-libtermcap.i686
yum install -y libidn.i686
yum install -y libSM.i686
yum install -y libuuid.i686
```

If the correct library is unavailable, CA Single Sign-On displays the following error:

```
java.lang.UnsatisfiedLinkError
```

## Linux Tools Required

Before installing a CA Single Sign-On Agent on a Red Hat Apache 2.2 web server running on the Red Hat Enterprise Linux operating environment, install all the items included in the Red Hat Legacy Software Development tools package.

## Compile an Apache Web Server on a Linux System

For the CA Single Sign-On Agent to operate with an Apache web server running Linux, you have to compile the server. Compiling is required because the Agent code uses pthreads (a library of POSIX-compliant thread routines), but the Apache server on the Linux platform does not, by default.

If you do not compile with the lpthreadoption, the Apache server starts up, but then hangs and does not handle any requests. The Apache server on Linux cannot initialize a module which uses pthreads due to issues with Linux's dynamic loader.

**Follow these steps:**

1. Enter the following:

```
LIBS=-lpthread
export LIBS
```

2. Configure Apache as usual by entering the following:

```
configure --enable-module=so --prefix=your_install_target_directory
make
make install
```

## Verify Presence of a Logs Subdirectory with Permissions for Apache-based CA Single Sign-On Agents

For agents running on Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

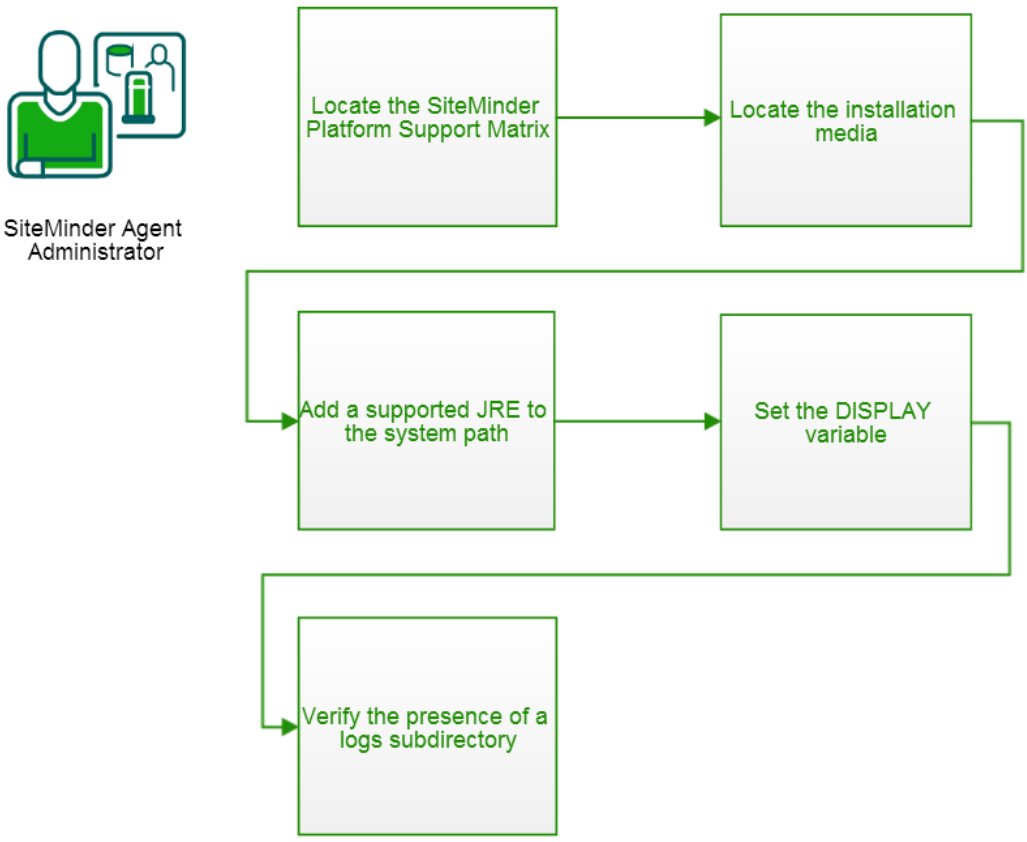
**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

# WSS Agent Preparations for z/OS

**Contents**



Before you install and configure an CA Single Sign-on WSS Agent on the z/OS operating environment, perform the preparation steps described in this process.




## Locate the Installation Media

To locate and download installation media, go to the CA Support site (https://support.ca.com/).

## Add a Supported JRE to the System Path

On z/OS systems, before installing the CA Single Sign-On agent, verify that a supported JRE is present on the system and defined in the PATH and JAVA_HOME system variables.

**Follow these steps:**

Enter the following commands at a command prompt:

```
export PATH=JRE/bin:$PATH
export JAVA_HOME=JRE
```

- ***JRE***
  Specifies the location of the JRE.
  For example, /sys/java64bt/v6r0m1/usr/lpp/java/J*version_number*

### Set the DISPLAY Variable for CA Single Sign-On Agent Installations on z/OS

If you are installing the CA Single Sign-On agent on a z/OS system from a remote terminal, verify that the DISPLAY variable is set for the local system. For example, if your server IP address is 111.11.1.12, set the variable as follows:

```
export DISPLAY=111.11.1.12:0.0
```

**Note:** You can also install the CA Single Sign-On agent using the console mode installation, which does not require the X window display mode.

### Verify Presence of a Logs Subdirectory with Permissions for Apache-based CA Single Sign-On Agents

For any agents for Apache-based web servers (including IBM HTTP Server), a logs subdirectory must exist under the root directory of the Apache-based web server. This subdirectory needs Read and Write permissions for the user identity under which the Apache child process runs.

If the logs subdirectory does not exist, create it with the required permissions.

**Note:** This configuration requirement applies to any Apache-based web server that writes log files outside the Apache root directory.

# IBM HTTP Server Preparations for WSS Agents

### Enable Write Permissions for IBM HTTP Server Logs

If you install the CA Single Sign-On Agent on an IBM HTTP Server, this web server gets installed as root and its subdirectories do not give all users in all groups Write permissions.

For the Low Level Agent Worker Process (LLAWP) to write agent initialization messages to the web server logs, the user running the web server needs permission to write to the web server's log directory. Ensure that you allow write permissions for this user.

# Install and Configure WSS Agents for Apache-based Servers on Windows

**Contents**

## Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.
2. Double-click System.
3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:

- *JVM_HOME*\jre\lib\security (Windows)
- *JVM_HOME*/jre/lib/security (UNIX)

  *JVM_HOME* is the installed location of the JVM used by the application server.

2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.

3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Gather the Information for the Installation Program

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path

## Run the Installer to Install a WSS Agent

Install the WSS Agent using the Web Services Security installation media on the Technical Support site.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the installation material.
3. Double-click ca-sm-wss-version-*cr*-win32.exe.

- ***cr***
  Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

The Web Services Security installation wizard starts.

> **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

4. Use gathered system and component information to install the WSS Agent. Consider the following points when running the installer:
   - When prompted to select which Web Services Security Agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - If the installer detects the presence of an existing Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.
5. Review the information that is presented on the Pre-Installation Summary page, then click Install.

   > **Note:** If the installation program detects that newer versions of certain system DLLs are installed on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location.
6. On the Web Services Security Configuration screen, click one of the following options and click Next:
   - Yes. I would like to configure Web Services Security Agents now.
   - No. I will configure Web Services Security Agents later.

If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

7. Click Done.
   If you selected the option to configure WSS Agents now, the installation program prepares the Web Services Security Configuration Wizard and begins the trusted host registration and configuration process. Use the information that you gathered earlier to complete the wizard. If you did not select the option to configure WSS Agents now, or if you are required to reboot the system after installation, run the configuration wizard manually later.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log

  - ***WSS_Home***
    Specifies the path to where Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security

  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.

- The Agent cannotcommunicate properly with the Policy Server until the trusted host is registered.

## Gather Information Required for WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

**Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed: Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like: policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Run the WSS Agent Configuration Program on Windows

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:

   *WSS_Home*\install_config_info

- ***WSS_Home***
  Specifies the path to where CA Single Sign-On Web Services Security is installed.
  **Default:** C:\Program Files\CA\Web Services Security

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, right-click ca-pep-config.exe, and then select Run as Administrator:

   - For a console-based configuration, enter the following command from a Command Prompt window with Administrator privileges open to *WSS_Home*\install_config_info:

     ```
     ca-pep-config.exe -i console
     ```

     **Important!** If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

3. Use the information you gathered earlier to complete the wizard.
   The agent runtime instance is created for your web servers.

## (Optional) Run an Unattended Installation and Configuration for Additional WSS Agents

The unattended installation option allows you to install and configure additional WSS Agents after the initial Agent is set up. It automates set up because it requires no user intervention. If you have a large Web Services Security environment that uses many agents with identical settings, using an unattended installation and configuration saves time.

To run an unattended setup, use the installation wizard or console-based installation program for your first installation. Afterwards, create your own script to run the installation program with the . properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):

   a. The Web Services Security Installation wizard.

   b. The Web Services Security Configuration wizard.

2. Locate the following file on your first web server:

   *WSS_Home*\install_config_info\ca-wss-installer.properties

   **Note**: If the path contains spaces, surround it with quotes.

- **WSS_Home**
  Specifies the path to where Web Services Security is installed.
  Default: C:\Program Files\CA\Web Services Security

3. Perform each of the following steps on the other web servers in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.

   b. Copy the following files from your first web server (from Steps 1 and 2) to the temporary directory on your subsequent web server:

      - The WSS Agent Installation executable file.
      - The ca-pepconfig-installer.properties file.

   c. Open a Command Prompt window with Administrative privileges in the temporary directory.

   d. Run the following command:

      ```
      ca-sm-wss-version-cr-win32.exe -f properties_file -i silent.
      ```

      - ***cr***
        Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

      **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

      The WSS Agent is installed and configured on the subsequent server automatically.

   e. (Optional) Delete the temporary directory from your subsequent web server.

4. Repeat Step 3 for each additional web server in your environment that uses the configuration that the settings in your ca-wss-installer.properties file specify.

# Install and Configure WSS Agents for Apache-based Servers on UNIX/Linux

**Contents**



## Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

**To set the JRE in the PATH variable**

1. Open a Command Window.
2. Run the following commands:

```
PATH=$PATH:JRE
export PATH
```

   - ***JRE***
     Defines the location of your Java Runtime Environment bin directory.

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.
3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Run the Installer to Install a CA Single Sign-on WSS Agent Using a GUI

Install the CA Single Sign-on WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-version-cr-unix_version.bin
  ```

  - ***cr***
    Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

  - ***unix_version***
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**To install the CA Single Sign-on WSS Agent**

1. Exit all applications that are running.

2. Open a shell and navigate to where the install program is located.

3. Enter the following command:

   ```
   ./ca-sm-wss-version-cr-unix_version.bin
   ```

   The CA Single Sign-On Web Services Security installer starts.

4. Use gathered system and component information to install the CA Single Sign-on WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**

   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

- If the installer detects the presence of an existing CA Single Sign-On Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a CA Single Sign-on WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

- Do *not* use spaces in the CA Single Sign-on WSS Agent install path.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The CA Single Sign-on WSS Agent files are copied to the specified location. Afterward, the CA CA Single Sign-On Web Services Security Configuration screen is displayed.

6. Select one of the following options:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.

   - No. I will configure CA Single Sign-On Web Services Security Agents later.

7. Click Done.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.

  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.

  - ***install-date-and-time***
    Specifies the date and time that the CA Single Sign-on WSS Agent was installed.

- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Run the Installer to Install a CA Single Sign-on WSS Agent Using a UNIX Console

Install the CA Single Sign-on WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-version-cr-unix_version.bin
  ```

  - ***cr***
    Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

  - ***unix_version***
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**To install the CA Single Sign-on WSS Agent**

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.
3. Enter the following command:

   ```
   ./ca-sm-wss-version-cr-unix_version.bin -i console
   ```

   The CA Single Sign-On Web Services Security installer starts.

4. Use gathered system and component information to install the CA Single Sign-on WSS Agent. Consider the following as you make your selections:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agentfor Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - Do not uses spaces in the CA Single Sign-on WSS Agent install path.
   - If the installer detects the presence of an existing CA Single Sign-On Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a CA Single Sign-on WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

5. Review the information presented on the Pre-Installation Summary page, then proceed.

   Note: If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

The CA Single Sign-on WSS Agent files are copied to the specified location. Afterward, the CA CA Single Sign-On Web Services Security Configuration screen is displayed.

6. Select one of the following options:
   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.
7. Hit Enter.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the CA Single Sign-on WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Gather Information Required for CA Single Sign-on WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .
- **SM Admin Password**
  The Policy Server administrator account password.
- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

**Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Set Environment Variables for a CA Single Sign-on WSS Agent on UNIX

After installing the CA Single Sign-on WSS Agent on UNIX, you must set required environment variables using the ca_wa_env.sh script. Running the script for CA Single Sign-on WSS Agents on most UNIX platforms ensures that the CA Single Sign-on WSS Agent and web server can work together.

The ca_wa_env.sh script sets the following environment variables:

- NETE_WA_ROOT
- PATH
- NETE_WA_PATH
- LD_LIBRARY_PATH

  **Note:** The CA Single Sign-on WSS Agent requires that LD_LIBRARY_PATH include /usr/lib before any other directory containing older versions of libm.so.

- SHLIB_PATH
- LIBPATH

To set the CA Single Sign-on WSS Agent environment variables after installation, source the following script after you install and configure the CA Single Sign-on WSS Agent:

1. Open a command window.
2. Navigate to *WSS_Home*/webagent/.

   - ***WSS_Home***
     Specifies the path to where CA Single Sign-On Web Services Security is installed.

3. Enter the following command:
   . ./ca_wa_env.sh

**Note:** You do not have to run this script for Sun Java System web servers because this file as been added to the start script.

## Run the CA Single Sign-on WSS Agent Configuration Program on UNIX or Linux Systems

You can configure your CA Single Sign-on WSS Agents and register a trusted host immediately after installing the CA Single Sign-on WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a CA Single Sign-on WSS Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.
- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

**To configure Agents and register a trusted host**

1. If necessary, start the Configuration Wizard as follows:
   a. Open a console window.
   b. Navigate to *agent_home*/install_config_info, where *agent_home* is the installed location of the CA Single Sign-on WSS Agent.
   c. Enter one of the following commands:
   GUI Mode: ./ca-pep-config.bin
   Console Mode: ./ca-pep-config.bin -i console

   The Configuration Wizard starts.
2. Use gathered system and component information to configure the CA Single Sign-on WSS Agent and register the host.

> **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*/config. You can modify this file.

- ***agent_home***
  Is the installed location of the CA Single Sign-on WSS Agent

## Set the LD_PRELOAD Variable

Most Apache-based CA Single Sign-On agents require that the LD_PRELOAD variable is set to the following value:

```
LD_PRELOAD=web_agent_home/bin/libbtunicode.so
```

> **Note**: Embedded Apache web servers included with RedHat Linux require different configuration procedures.

## Set the LD_ASSUME_KERNEL for Apache Agent on SuSE Linux 9 for zSeries

After you install the Web Agent on an Apache web server running on SuSE Linux 9 for zSeries, set the LD_ASSUME_KERNEL environment variable as follows:

LD_ASSUME_KERNEL=2.4.21

export LD_ASSUME_KERNEL

**Important!** You must set this variable to 2.4.21 because it represents the kernel release upon which the Web Agent libraries are built.

Without this setting, the following problems occur:

- The Apache web server will not start properly.
- Host registration dumps core.

## Set the CAPKIHOME Variable for Red Hat Linux 6 Systems

If you want to run an Apache-based Web Agent on a Red Hat Linux system, set the CAPKIHOME environment variable by entering the following commands:

```
CAPKIHOME="/usr/local/CA/webagent/CAPKI"
export CAPKIHOME
```

## (Optional) Run the Unattended or Silent Installation and Configuration Programs for your CA Single Sign-on WSS Agent

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On Web Services Security environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
    a. The CA Single Sign-On Web Services Security Installation wizard.
    b. The CA Single Sign-On Web Services Security Configuration wizard.
2. Locate the following file on your first web server:

    `WSS_Home/install_config_info/ca-wss-installer.properties`

> **Note**: If the path contains spaces, surround it with quotes.

- ***WSS_Home***
  Specifies the path to where CA Single Sign-On Web Services Security is installed.

3. Perform each of the following steps on the subsequent web servers:

> Note: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.

   b. Copy the following files from the web server where you ran the wizards (from Steps 1 and 2) to the temporary directory on your subsequent web server:

      - The CA Single Sign-on WSS Agent Installation executable file.
      - The ca-pepconfig-installer.properties file.

   c. Open a Command Prompt window with root privileges in the temporary directory.

   d. Run the following command:

      ```
      ca-sm-wss-version-cr-unix_version.bin -f properties_file -i silent
      ```

      - ***cr***
        Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

      The CA Single Sign-on WSS Agent is installed and configured on the web server silently.

   e. (Optional) Delete the temporary directory from your web server.

4. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wss-installer.properties file specify.

# Optional Agent Settings for UNIX/Linux on Apache-based Servers

**Contents**

## Set CA Single Sign-on WSS Agent Variables when using apachectl Script

You run your Apache server using the apachectl script (such as when running an Apache web server on POSIX). Add a line to the apachectl script to set the environment variables for the CA Single Sign-On agent.

**Follow these steps:**

1. Locate a line resembling the following example:

   ```
   # Source /etc/sysconfig/httpd for $HTTPD setting, etc
   ```

2. Add the following line *before* the line in the previous example:

   ```
   sh /agent_home/ca_wa_env.sh
   ```

   - ***agent_home***
     Specifies the installed location of the CA Single Sign-on WSS Agent.

## Improve Server Performance with Optional httpd.conf File Changes

You can improve server performance by modifying the default configuration settings in the httpd.conf file; however, these changes are *not* required:

**Follow these steps:**

1. For Apache- based servers, assign a higher priority level to your Apache20WebAgent.dll file than any other auth or access modules on your web server.
2. For low-traffic websites, define the following directives:
   - Set MaxRequestsPerChild>1000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >5
   - MaxSpareServers>10
   - StartServers=MinSpareServers>5
3. For high-traffic websites, define the following directives:
   - Set MaxRequestsPerChild>3000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >10
   - MaxSpareServers>15
   - StartServers=MinSpareServers>10

**Note:** CA Services can provide assistance with performance-tuning for your particular environment.

# Install and Configure WSS Agents for Apache-based Serves on z/OS Systems

**Contents**



## Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Run the CA Single Sign-on WSS Agent Installation Program on z/OS

The installation program for the CA Single Sign-On agent installs the agent on a single computer running the z/OS operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

You install the CA Single Sign-On agent using the installation media on the Technical Support site.

**Note:** Verify that you have executable permissions. To add executable permissions to the installation media, run the following command:

```
chmod +x installation_media
```

- ***installation_media***
  Specifies the CA Single Sign-on WSS Agent installer executable.

**Follow these steps:**

1. Log in as a root user.
2. Exit all applications that are running.
3. Open a shell and navigate to the installation media.
4. Run the installation program in GUI or console mode by entering one of the following commands:
   GUI Mode:

   ```
   java -jar  installation_media
   ```

   Console Mode:

   ```
   java -jar installation_media -i console
   ```
5. Use gathered system and component information to install the CA Single Sign-on WSS Agent. Consider the following when running the installer:
   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - Do not use space characters in the CA Single Sign-on WSS Agent install path. For example, " /CA Technologies/agent" will result in install failure.
   - If the installer detects the presence of an existing CA Single Sign-On Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a CA Single Sign-on WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.
6. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The CA Single Sign-on WSS Agent files are copied to the specified location. Afterward, the CA CA Single Sign-On Web Services Security Configuration screen is displayed.
7. Select one of the following options:
   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

8. Click Done.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the CA Single Sign-on WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Gather Information Required for CA Single Sign-on WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .
- **SM Admin Password**
  The Policy Server administrator account password.
- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.
- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

> **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Set the Library Path Variable on z/OS

Set the library path variable on z/OS systems before running the agent configuration program.

```
export LIBPATH=agent_home/bin
```

***agent_home***

Indicates the directory where the CA Single Sign-on WSS Agent is installed on your web server.

**Default** (Windows 32-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent

**Default** (Windows 64-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64

**Default** (Windows 32-bit CA Single Sign-on WSS Agent installations operating on 64-bit systems: C:\Program Files(x86)\CA\Web Services Security\webagent\win32

## Run the CA Single Sign-on WSS Agent Configuration Program on z/OS

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other z/OS systems in the future.

**Note:** Verify that you have executable permissions. To add executable permissions to the installation media, run the following command:

```
chmod +x installation_media
```

- ***installation_media***
  Specifies the CA Single Sign-On agent installer executable.

**Follow these steps:**

1. Log in as a root user.
2. Exit all applications that are running.
3. Open a shell and navigate to the following directory:

   ```
   WSS_home/install_config_info
   ```

   - ***WSS_home***
     Specifies the installed location of the <soasm).
4. Run the configuration program in GUI or console mode by entering one of the following commands:
   GUI Mode:

   ```
   ca-wa-config.sh
   ```

   Console Mode:

   ```
   ca-wa-config.sh -i console
   ```
5. Follow the prompts shown in the configuration program. Provide the requested values from your agent configuration worksheet.
   The agent runtime instance is created for your web servers.

## (Optional) Run the Unattended or Silent Installation and Configuration Programs for CA Single Sign-on WSS Agents on z/OS

The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first web server (in the order shown):
   a. The CA Single Sign-on WSS Agent Installation wizard.
   b. The CA Single Sign-on WSS Agent configuration wizard.
2. Locate the following file on your first web server:

   ```
   WSS_Home\install_config_info\ca-wss-installer.properties
   ```

   **Note**: If the path contains spaces, surround it with quotes.

   - ***WSS_Home***
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
3. Perform each of the following steps on the subsequent web server:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on the subsequent web server.
   b. Copy the following files from the web server where you ran the wizards (from Steps 1 and 2) to the temporary directory on your subsequent web server:
      - The CA Single Sign-on WSS Agent Installation executable file.
      - The ca-pepconfig-installer.properties file.
   c. Open a Command Prompt window with root privileges in the temporary directory.
   d. Run the following command:

      ```
      java -jar installation_media -f ca-pepconfig-installer.properties -
      i silent
      ```

      - ***installation_media***
        Specifies the CA Single Sign-on WSS Agent installer executable.

      The CA Single Sign-on WSS Agent is installed and configured on the web server silently.

e. (Optional) Delete the temporary directory from your web server.

4. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-pepconfig-installer.properties file specify.

# WSS Agent Settings for Apache-based Servers

**Contents**



## Use the HttpsPorts Parameter on Apache 2.x Servers

More web server configuration is required of all of the following conditions exist:

You use an SSL accelerator or any intermediate device that changes the value of the HTTP_HOST header with an Apache 2.x Web server.

- You use the HttpsPorts parameter.

**Follow these steps:**

1. Open the httpd.conf file of your Apache Web server, and then make the following changes:
   - Change the value of the UseCanonicalName parameter to on.
   - Change the value of the ServerName parameter to the following format:

     ```
     server_name:port_number
     ```

     - ***server_name***
       Specifies the host name of the SSL accelerator.
2. For your Web Agent, change the value of the GetPortFromHeaders parameter to yes.

## Use Legacy Applications with an Apache Web Agent

If you have legacy applications (that do not support HTTP 1.1), and you want to run them on an Apache Web Server, you can set the following parameter:

- **LegacyTransferEncodingBehavior**
  Specifies the type of message encoding used by the Web Agent. When the value of this parameter is set to no, transfer-encoding is supported.
  When the value of this parameter is set to yes, content encoding is used. The transfer-encoding header is ignored and only the content-length header is supported.
  **Default**: No

To use legacy applications with an Apache Web Server, set the value of the LegacyTransferEncodingBehavior parameter to yes.

**Important!** If you set the value of this parameter to yes, these features will not work: Federation; preservation of POST data longer than 4 KB; and large certificates may not be recognized.

## Use the HTTP HOST Request for the Port Number

If you have applications that perform load balancing by redirecting traffic to specific web servers *without* modifying the actual HTTP headers, you should configure the Web Agent to redirect users back to the proper external port (instead of the port used by the load balancer) with the following parameter:

- **GetPortFromHeaders**
  Directs the Web Agent to obtain the port number from the HTTP HOST request header instead of obtaining it from the web server service structures.
  **Default**: No
  **Note**: This parameter is required for Apache Web Agents.

## Record the Transaction ID in Apache Web Server Logs

The Web Agent generates a unique transaction ID for each successful user authorization request. The Agent adds the ID to the HTTP header. The ID is also recorded in the following logs:

- Audit log
- Web server log (if the server is configured to log query strings)
- Policy Server log

You can track user activities for a given application using the transaction ID.

Note: For more information, see the Policy Server documentation.

The transaction ID appears in the log as a mock query parameter in the log that is appended to the end of an existing query string. The following example shows transaction ID (in bold) appended to a query string (which ends with STATE=MA):

```
172.24.12.1, user1, 2/11/00, 15:30:
10, W3SVC, MYSERVER, 192.168.100.100, 26844, 47, 101, 400, 123, GET, /realm/index.
html, STATE=MA&SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1
```

If no query parameters are in the URL, the Agent adds the transaction ID at the end of the web server log entry. For example:

```
172.24.12.1, user1, 2/11/00, 15:30:
10, W3SVC, MYSERVER, 192.168.100.100, 26844, 47, 101, 400, 123, GET, /realma/index.
html, SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1.
```

**Note:** Web Agents log user names and access information in native web server log files when users access resources.

You can record the CA Single Sign-On transaction ID in the Apache web server logs SMTRANSACTIONID header variable.

**Follow these steps:**

1. Open the httpd.conf file.
2. Add the SM_TRANSACTIONID header variable to the LogFormat directive.
   For example:

   ```
   LogFormat "%h %l %u %t \"%r\" %>s %b \"%{SM_TRANSACTIONID}i\"" common
   ```

   **Note:** For more information about the httpd.conf file and the LogFormat directive, see your Apache web server documentation.

3. Restart the server to apply the change.
   The transaction ID is recorded in the Apache web server logs.

## Choose How Content Types are Transferred in POST Requests

If you are using an Apache web server, you can control how content is transferred to the server during POST requests with the following parameter:

LegacyStreamingBehavior

Specifies how content will be transferred to the server during POST requests. When the value of this parameter is set to yes, all content types are streamed, *except* for the following:

- text/xml
- application/x-www-form-urlencoded

When the value of this parameter is set to no, all content types are spooled.

**Default**: No

To stream most types of content in POST requests, change the value of the LegacyStreamingBehavior parameter to yes.

## Restrict IPC Semaphore-Related Message Output to the Apache Error Log

By default the Apache Web Agent logs all levels (informational and error) of IPC semaphore-related messages to the Apache error log, regardless of the configured Apache logging level.

To restrict the verbosity of Web Agent IPC semaphore-related output to the Apache error log, add the following parameter in the trace.conf file located in *web _agent_home*/config:

- **nete.stderr.loglevel**
  Specifies the level of IPC semaphore-related messages the Web Agent logs to the Apache error log. Accepts the following values:
  - **off**
    The Web Agent logs no IPC semaphore-related messages to the Apache error log.
  - **error**
    The Web Agent logs only IPC semaphore-related error messages to the Apache error log.
  - **info**
    (Default) The Web Agent logs IPC semaphore-related error and informational messages to the Apache error log.

**Example: Define the nete.stderr.loglevel parameter in trace.conf**

In the following snippet from trace.conf, the nete.stderr.loglevel parameter is configured to restrict the Web Agent to log only IPC semaphore-related *error* messages to the Apache error log:

```
# CA Web Agent IPC logging levels
# nete.stderr.loglevel=error
```

## Delete Certificates from Stronghold (Apache Agent Only)

Stronghold web servers write client certificates to a local, temporary file, which the Web Agent uses for certificate-based authentication. The Stronghold server uses this file to make information in the client certificate available for authentication. As users visit a website, these certificate files increase, taking up space on your server. You can configure the Web Agent to delete a certificate file after the Agent has finished using it.

To delete certificate files, set the DeleteCerts parameter to yes.

# Uninstall a SiteMinder WSS Agent from Apache-based Servers

**Contents**

- Uninstall a CA Single Sign-on WSS Agent (see page 673)

## Set JRE in PATH Variable Before Uninstalling the CA Single Sign-On Agent

On Windows and UNIX systems, when you are uninstalling a CA Single Sign-On Agent, make sure the JRE is in the PATH variable or the uninstallation program stops and issues one of the following error messages:

- “Could not find a valid Java virtual machine to load. You need to reinstall a supported Java virtual machine.”
- "No Java virtual machine could be found from your PATH environment variable. You must install a VM prior to running this program."

**Follow these steps:**

**On Windows**

1. Go to the Control Panel.
2. Double-click System.
3. In the Environment Variables dialog, add the location of the JRE to the PATH system variable. For example, C:\j2sdk*version_number*\jre\bin

**On UNIX**

Run the following commands:

1. PATH=$PATH:*JRE*/bin
   - ***JRE***
     Specifies the location of your JRE.
     For example, /usr/bin/j2sdk*version_number*/jre
2. export PATH

## Uninstall a CA Single Sign-on WSS Agent

To uninstall a CA Single Sign-on WSS Agent, run the CA Single Sign-On Web Services Security uninstall wizard.

**Follow these steps:**

1. Navigate to the WSS_*HOME*\install_config_info (Windows) or WSS_*HOME*/install_config_info (UNIX) directory and run the CA Single Sign-On Web Services Security uninstall wizard to remove core CA Single Sign-On Web Services Security components:
   - Windows: wss-uninstall.cmd
   - UNIX: wss-uninstall.sh

- ***WSS_HOME***
  Specifies the CA Single Sign-On Web Services Security installation location.

> **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.

3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.
   The uninstall wizard removes all selected CA Single Sign-On Web Services Security components.

4. Restart the server.

# SiteMinder WSS Agent Logging for Apache-based Servers

**Contents**



## Logs of Start-up Events

To assist in debugging, startup events are recorded in a log. Each message may provide clues about the problem. These logs are stored in the following locations:

- On Windows systems, these events are recorded in the Windows Application Event log.

- On UNIX systems, these events are sent to STDERR. Apache servers map STDERR to the Apache error_log file, so these events are also recorded in that log.

## How to Set Up Trace Logging

To set up trace logging, use the following process:

1. Set up and Enable Trace logging.

2. Determine what you want to record in the trace log by reviewing the following lists:

- Trace Log Components and Subcomponents
- Trace Message Data Fields
- Data Field Filters

3. Duplicate the default Trace Configuration File.
4. Modify the duplicate file to include the items you want to record.
5. Restart the agent.

## Configure Trace Logging

Before you can use trace logging, you must configure it by specifying a name, location, and parameters for the trace log file. These settings control the size and format of the file itself. After trace logging is configured, you determine the content of the trace log file separately. This lets you change the types of information contained in your trace log at any time, without changing the parameters of the trace log file itself.

**Follow these steps:**

1. Locate the WebAgentTrace.conf file on your web server. Duplicate the file.
2. Open your Agent Configuration Object or local configuration file.
3. Set the TraceFile parameter to yes.

**Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig. conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.

4. Specify the full path to the trace log files in following parameters:
   - **TraceFileName**
     Specifies the full path to the trace log file.
     Default: No default
     Limits: Specify the file name in this parameter.Example: web_agent_home\log\trace.log
5. Specify the full path to the duplicate copies of WebAgentTrace.conf file (you created in Step 1) in the following parameters:
   - **TraceConfigFile**
     Specifies the location of the WebAgentTrace.conf configuration file that determines which components and events to monitor.

**Default**: No default
**Example**: web_agent_home\config\WebAgentTrace.conf
**Note:** This file is not used until the web server is restarted.

6. Define the format of the information in your trace log file by setting the following parameters in your Agent Configuration Object or local configuration file:

   - **TraceAppend**
     Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
     **Default**: No

   - **TraceFormat**
     Specifies how the trace file displays the messages. Choose one of the following options:

     - default—uses square brackets [] to enclose the fields.
     - fixed—uses fields with a fixed width.
     - delim—uses a character of your choice to delimit the fields.
     - xml—uses XML-like tags. A DTD or style sheet is not provided with the Web Agent.

     **Default**: default (square brackets)

   - **TraceDelimiter**
     Specifies a custom character that separates the fields in the trace file.
     **Default**: No default
     **Example**: |

   - **TraceFileSize**
     Specifies (in megabytes) the maximum size of a trace file. The Web Agent creates a new file when this limit is reached.
     **Default:** 0 (a new log file is not created)
     **Example**: 20 (MB)

   - **LogLocalTime**
     Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
     **Default**: Yes

7. Edit the WebAgentTrace.conf file to have agent monitor the activities you want.
   Framework Agents do not support dynamic configuration of log parameters set locally in the Agent configuration file. Consequently, when you modify a parameter, the change does not take effect until you restart the web server. However, these log settings can be stored and updated dynamically if you configure them in an Agent configuration object on the Policy Server.

8. Restart the web server so the agent uses the new trace configuration file.

## Trace Log Components and Subcomponents

The CA Single Sign-On Agent can monitor specific CA Single Sign-On components. When you monitor a component, all of the events for that component are recorded in the trace log. Each component has one or more subcomponents that the agent can also monitor. If you do not want the agent to record all of the events for a component, you can specify only those subcomponents you want to monitor instead.

For example, if you want to record only the single sign-on messages for an agent on a web server, you would specify the WebAgent component and the SSO subcomponent.

The following components and subcomponents are available:

- **AgentFramework**
  Records all Agent framework messages. (Applies only to framework agents.) The following subcomponents are available:
  - Administration
  - Filter
  - HighLevelAgent
  - LowLevelAgent
  - LowLevelAgentWP
- **AffiliateAgent**
  Records web Agent messages related to the 4.x Affiliate Agent, which is part of Federation Security Services, a separately-purchased product. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **SAMLAgent**
  Web Agent messages related to the SAML Affiliate Agent. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **WebAgent**
  Records all Web Agent log messages. Applies to all Agents *except* IIS 6.0 or Apache 2.0 Agents. The following subcomponents are available:
  - AgentCore
  - Cache
  - authentication
  - Responses

- Management
- SSO
- Filter

- **Agent_Functions**
  Records all Agent API messages. The following subcomponents are available:
  - Init
  - UnInit
  - IsProtected
  - Login
  - ChangePassword
  - Validate
  - Logout
  - Authorize
  - Audit
  - FreeAttributes
  - UpdateAttributes
  - GetSessionVariables
  - SetSessionVariables
  - DeleteSessionVariables
  - Tunnel
  - GetConfig
  - DoManagement
- **Agent_Con_Manager**
  Records messages related to internal processing of the Agent API. The following subcomponents are available:
  - RequestHandler
  - Cluster
  - Server

- WaitQueue
- Management
- Statistics

## Trace Message Data Fields

You can define what each trace message for a specific component contains by specifying which data fields to include in the message.

Data fields use the following syntax:

```
data:data_field1,data_field2,data_field3
```

Some data fields are shown in the following example:

```
data:message,date,time,user,agentname,IPAddr
```

There may not be data for fields in each message, so blank fields my occur. For example, if you select RealmOID as a data field, some trace messages will display the realm's OID while others will not.

The following data fields are available:

- **Message**
  Includes the actual trace message
- **SrcFile**
  Includes the source file and line number of the trace message
- **Pid**
  Includes the process ID
- **Tid**
  Includes the thread ID
- **Date**
  Includes the date
- **Time**
  Includes the time
- **PreciseTime**
  Includes the time, including milliseconds
- **Function**
  Includes the function in the code containing the trace message
- **User**
  Includes the name of the user
- **Domain**
  Includes the CA Single Sign-On domain

- **Realm**
  Includes the CA Single Sign-On realm
- **AgentName**
  Includes the Agent name being used
- **TransactionID**
  Includes the transaction ID
- **DomainOID**
  Includes the CA Single Sign-On domain OID
- **IPAddr**
  Includes the client IP address
- **RequestIPAddr**
  Includes the trace file displays the IP of the server where Agent is present
- **IPPort**
  Includes the client IP port
- **CertSerial**
  Includes the certificate serial number
- **SubjectDN**
  Includes the subject DN of the certificate
- **IssuerDN**
  Includes the Issuer DN of the certificate
- **SessionSpec**
  Includes the CA Single Sign-On session spec
- **SessionID**
  Includes the CA Single Sign-On session ID
- **UserDN**
  Includes the User DN
- **Resource**
  Includes the requested resource
- **Action**
  Includes the requested action
- **RealmOID**
  Includes the realm OID
- **ResponseTime**
  Includes the average response time in milliseconds of the Policy Servers associated with a CA Web Agent or SDK Agent and API application
  **Note:** To output the ResponseTime to a trace log, include the component Agent_Con_Manager

along with the data field ResponseTime in the WebAgentTrace.conf file or other file specified in the Policy Server Configuration Object (ACO) and restart the Policy Server. The Agent_Con_Manager component, or Agent API Connection Manager, calculates the ResponseTime each time a response is received from a Policy Server and keeps a running average. To locate the ResponseTime in the trace log, search for [PrintStats].

## Trace Message Data Field Filters

To focus on a specific problem, you can narrow the output of the trace log by specifying a filter based on the value of a data field. For example, if you are having problems with an index.html page, you can filter on resources with an html suffix by specifying Resource:==/html in the trace configuration file. Each filter should be on a separate line in the file.

Filters use the following syntax:

*data_field*:*filter*

The following types of filters are available:

- == (exact match)
- != (does not equal)

The filters use boolean logic as shown in the following examples:

Action:!=get (all actions except get)

Resource:==/html (all resources ending in /html)

## Determine the Content of the Trace Log

The WebAgentTrace.conf file determines the content of the trace log. You can control which components and data items appear in your trace log by modifying the settings of the WebAgentTrace.conf file on your web server. The following factors apply when editing the file:

- Entries are case-sensitive.
  When you specify a component, data field, or filter, the values must match exactly the options in the WebAgentTrace.conf file instructions.
- Uncomment the configuration settings lines.
- If you modify the WebAgentTrace.conf file before installing a new agent over an existing agent, the file is overwritten. Rename or back up the file first. After the installation, you can integrate your changes into the new file.

**Follow these steps:**

1. Open the WebAgentTrace.conf file.

> **Note:** We recommend duplicating the original file and changing the copy. Modifying the copy preserves the default settings.

2. Add components and subcomponents using the following steps:

    a. Find the section that matches your type of agent. For example, if you have an Apache 2.0 Agent that is installed on your server, look for a line resembling the following example:

    ```
    # For Apache 2.0, Apache 2.2, IIS 7.0 and SunOne Web Agents
    ```

    b. Locate the following line in that section:

    ```
    #components:
    ```

    c. Uncomment the line. Then add the component names that you want after the colon. Separate multiple components commas as shown in the following example:

    ```
    components:  AgentFramework, HTTPAgent
    ```

    d. (Optional) Follow the component name with the name of a subcomponent you want. Separate the subcomponent name with a slash as shown in the following example:

    ```
    components:  AgentFramework/Administration
    ```

3. Add data fields and filters using the following steps:

    a. Locate the following line in the appropriate section:

    ```
    #data:
    ```

    b. Uncomment the line. Then add the data fields that you want after the colon. Separate multiple data fields with commas as shown in the following example:

    ```
    data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr
    ```

    c. (Optional) Add filters to your data fields by following the data field with a colon, the Boolean operator and the value you want. The values you specify for the filters must match exactly. The following example shows a filter which logs activities for a specific IP address:

    ```
    data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr:
    ==127.0.0.1
    ```

    **Note:** Each filter must be on a separate line in the file.

4. Save your changes and close the file.

5. Restart the web server to apply your changes.
   The content of the trace log has been determined.

## Limit the Number of Trace Log Files Saved

You can limit the number of trace logs that a CA Single Sign-On agent keeps. For example, if you want to save disk space on the system that stores your agent logs, you can limit the number of trace logs using the following parameter:

TraceFilesToKeep

Specifies the number of CA Single Sign-On agent trace log files that are kept. New trace logs are created in the following situations:

- When the agent starts.
- When the size limit of the trace log (specified by the value of the TraceFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing trace logs which exceed the number that you want to keep. For example, If your system has 500 trace logs stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 trace logs.

Setting the value of this parameter to zero retains all the trace logs.

**Default**: 0

**Follow these steps:**

1. Archive or delete any existing trace logs from your system.
2. Set the value of the TraceAppend parameter to no.
3. Change the value of the TraceFilesToKeep parameter to the number of trace logs that you want to keep.

## Collect Detailed Agent Connection Data with an Agent Connection Manager Trace Log

To collect detailed information about the connections between a CA Single Sign-on WSS Agent and Policy Server, you create a Trace Log file that contains information gathered by the Agent Collection Manager.

**Follow these steps:**

1. Open your Agent Configuration object or local configuration file.
2. Set the value of the TraceFile parameter to yes.

**Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings defined on the Policy Server. For example, when the value of this parameter is set to yes in a LocalConfig.conf file log files are generated even if the value of the AllowLocalConfig parameter in the corresponding Agent Configuration object on the Policy Server is set to no. Additionally, set the related trace logging parameters (that define the file name, size, and so on) in the LocalConfig.conf file to override any Policy Server trace log settings.

3. Specify the full path to the trace log file for your Agent Connection Data in the TraceFileName parameter. This is the file that contains the trace log output.

4. Set the value of the TraceConfigFile parameter to the full path of the following file:

   `agent_home/config/AgentConMgr.conf`

   - ***agent_home***
     Indicates the directory where the CA Single Sign-on WSS Agent is installed on your web server.
     **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
     **Default** (Windows 64-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
     **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations operating on 64-bit systems: C:\Program Files(x86)\CA\Web Services Security\webagent\win32

5. Define the format the trace log file for your Agent Connection Data by setting the following parameters:

   - **TraceAppend**
     Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
     **Default:** No

   - **TraceDelimiter**
     Specifies a custom character that separates the fields in the trace file.
     **Default:** No default
     **Example:** |

   - **TraceFileSize**
     Specifies (in megabytes) the maximum size of a trace file. The Web Agent creates a new file when this limit is reached.
     **Default:** 0 (a new log file is not created)
     **Example:** 20 (MB)

   - **TraceFormat**
     Specifies how the trace file displays the messages. Choose *one* of the following options:

     - default—uses square brackets [] to enclose the fields.
     - fixed—uses fields with a fixed width.
     - delim—uses a character of your choice to delimit the fields.
     - xml—uses XML-like tags. A DTD or style sheet is *not* provided with the Web Agent.

     **Default:** default (square brackets)

   - **LogLocalTime**
     Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
     **Default:** Yes

6. Restart your web server so the new settings take effect.
   Detailed information about the CA Single Sign-on WSS Agent connections is collected.

> **Note**: For CA Single Sign-On 12.52, the BusyHandleCount and FreeHandleCount attributes are not used.

## Configure XML Message Processing Logging

In addition to Web Agent logging functionality, the CA Single Sign-on WSS Agent provides an additional level of log information relating specifically to its processing of XML messages. CA Single Sign-on WSS Agent logging is implemented using Apache's *log4j* standard (see http://logging.apache.org).

> **Note:** CA Single Sign-on WSS Agent logging does not start until an XML message that needs to be processed is received.

By default, CA Single Sign-on WSS Agent logging is enabled and written to the soasm_agent.log file in:

- Windows—*agent_home*\bin\
- UNIX—*agent_home*/bin/

  - ***agent_home***
    Indicates the directory where the CA Single Sign-on WSS Agent is installed on your web server.
    **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
    **Default** (Windows 64-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
    **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations operating on 64-bit systems: C:\Program Files(x86)\CA\Web Services Security\webagent\win32

You can change logging parameters for your CA Single Sign-on WSS Agent by editing the log.config file, which can be found in:

- Windows—*agent_home*\config\
- UNIX— *agent_home*/config/

## Disable CA Single Sign-on WSS Agent XML Message Processing Logging

To disable CA Single Sign-on WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent config subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

## Error Logs and Trace Logs

You can use the Web Agent logging function to monitor the performance of the Web Agent and its communication with the Policy Server. The logging feature provides accurate and comprehensive information about the operation of CA Single Sign-On processes to analyze performance and troubleshoot issues.

A log is a record of events that occur during program execution. A log consists of a series of log messages, each one describing some event that occurred during program execution. Log messages are written to log files.

**Note**: IIS Agents create log files only after the first user request is submitted. Apache 2.0 Web Agents create log files when the Apache server starts.

The Web Agent uses the following log files:

- **Error log**
  Contains program and operational-level errors. One example is when the Web Agent cannot communicate with Policy Server. The level of detail output in this log cannot be customized. Error logs contain the following types of messages:
  - **Error messages**
    Contain program-level errors, which indicate incorrect or abnormal program behavior, or an inability to function as expected due to some external problem, such as a network failure. There are also operational-level errors. This type of error is a failure that prevents the operation from succeeding, such as opening a file or authenticating a user.
  - **Informational messages**
    Contain messages for the user or administrator that some event has occurred; that is, that a server has started or stopped, or that some action has been taken.
  - **Warning messages**
    Contain warnings for the user or administrator of some condition or event that is unusual or indicative of a potential problem. This does not necessarily mean there is anything wrong.
- **Trace log**
  Contains detailed warning and informational messages, which you can configure. Examples include trace messages and flow state messages. This file also includes data such as header details and cookie variables. Trace logs contain the following messages:
  - **Trace messages**
    Provide detailed information about program operation for tracing and/or debugging purposes. Trace messages are ordinarily turned off during normal operation. In contrast to informational, warning, and error messages, trace messages are embedded in the source code and can not easily be localized. Moreover, trace messages may include significant data in addition to the message itself; for example, the name of the current user or realm.

You specify the location of both the error and trace log files when you configure the Web Agent. Use the error and trace logs to help solve any issues that may prevent the Web Agent from operating properly.

**Note:** For Agents on Windows platforms, set the EnableWebAgent parameter to yes to ensure that the Web Agent log gets created. If you leave EnableWebAgent set to no (the default) and set the logging parameters, the Agent log gets created only for Agents on UNIX platforms.

## Parameter Values Shown in Log Files

Web Agents list configuration parameters and their values in the Web Agent error log file, but there are differences between the ways that Traditional and Framework agents do this.

Framework agents record the configuration parameters and their values in the log file exactly as you entered them in the Agent Configuration Object or the local configuration file. All of the parameters, including those which may contain an incorrect value, are recorded in the log file.

Traditional agents process the parameter values before recording them. If the parameter has a proper value, the parameter and its value are recorded in the log file. Parameters with incorrect values are *not* recorded in the log file.

## Set Up and Enable Error Logging

Error logs require the following settings:

- Logging is enabled.
- A location for the log file is specified.

The parameters that enable error logging and determine options such as appending log data are defined in a local configuration file or an Agent Configuration Object at the Policy Server.

Agents that are installed on an IIS or Apache web servers do not support dynamic configuration of log parameters that are set locally in a local configuration file. The changes take effect when the Agent is restarts. However, these log settings can be stored and updated dynamically in an agent configuration object at the Policy Server.

**Note:** IIS Agents create log files only after the first user request is submitted. Apache 2.0 Web Agents create log files when the Apache server starts.

**Follow these steps:**

1. If you do not have a log file already, create a log file and any related directories.
2. Set the value of the LogFile parameter to yes.

**Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig.conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.

3. Specify the full path to the error file, including the file name, in any of the following parameters:

   - **LogFileName**
     Specifies the full path (including the file name) of the log file.
     **Default:** No
     **Example:** (Windows) *agent_home*\log\WebAgent.log
     **Example:** (UNIX/LInux) /export/iPlanet/servers/https-jsmith/logs/WebAgent.log

   - **LogFileName32**
     Specifies the full path of a log file for a CA Single Sign-on WSS Agent for IIS (on 64-bit Windows operating environments protecting 32-bit applications). The 32-bit applications run in Wow64 mode on the 64-bit Windows operating environment. If logging is enabled but this parameter is not set, the CA Single Sign-on WSS Agent for IIS appends _32 to the log file name.
     **Default**: No
     **Limits**: For Windows 64-bit operating environments only. Specify the file name at the end of the path.
     **Example**: (Windows 64-bit operating environments using Wow64 mode) *agent_home* \log\WebAgent32.log.

4. (Optional) Set the following parameters (in the Agent Configuration Object on the Policy Server or in the local configuration file):

   - **LogAppend**
     Adds new log information to the end of an existing log file. When this parameter is set to no, the entire log file is rewritten each time logging is invoked.
     **Default:** No

   - **LogFileSize**
     Specifies the size limit of the log file in megabytes. When the current log file reaches this limit, a new log file is created. The new log file uses one of the following naming conventions:

     - For framework agents, the new log file has a sequence number that is appended to the original name. For example, a log file named myfile.log is renamed to myfile.log.1 when the size limit is reached.

     - For traditional agents, the new log files are named by appending the date and timestamp to the original name. For example, a log file named myfile.log, is renamed to myfile.log.09-18-2003-16-07-07 when the size limit is reached.

Archive or remove the old files manually.
**Default:** 0 (no rollover)
**Example:** 80

- **LogLocalTime**
  Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
  **Default:** Yes
  If you use a local configuration file, your settings resemble the following example:

```
LogFile="yes"
LogFileName="/export/iPlanet/servers/https-myserver/logs/errors.log"
LogAppend="no"
LogFileSize="80"
LogLocalTime="yes"
```

Error logging is enabled.

## Enable Transport Layer Interface (TLI) Logging

When you want to examine the connections between the agent and the Policy Server, enable transport layer interface logging.

**To enable TLI logging**

1. Add the following environment variable to your web server.

   ```
   SM_TLI_LOG_FILE
   ```

2. Specify a directory and log file name for the value of the variable, as shown in the following example:

   ```
   directory_name/log_file_name.log
   ```

3. Verify that your agent is enabled.

4. Restart your web server.
   TLI logging is enabled.

## Limit the Number of Log Files Saved

You can limit the number of log files that an agent keeps. For example, if you want to save disk space on the system that stores your agent logs, you can limit the number of log files using the following parameter:

LogFilesToKeep

Specifies the number of agent log files that are kept. New log files are created in the following situations:

- When the agent starts.
- When the size limit of the log file (specified by the value of the LogFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing logs files which exceed the number that you want to keep. For example, If your system has 500 log files stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 files.

Setting the value of this parameter to zero retains all the log files.

**Default**: 0

**Follow these steps:**

Archive or delete any existing log files from your system.

Set the value of the LogAppend parameter to no.

Change the value of the LogFilesToKeep parameter to the number of log files that you want to keep.

# Web Services Security Agent for IIS Servers

The following sections detail how to install and configure a WSS agent on an IIS web server.

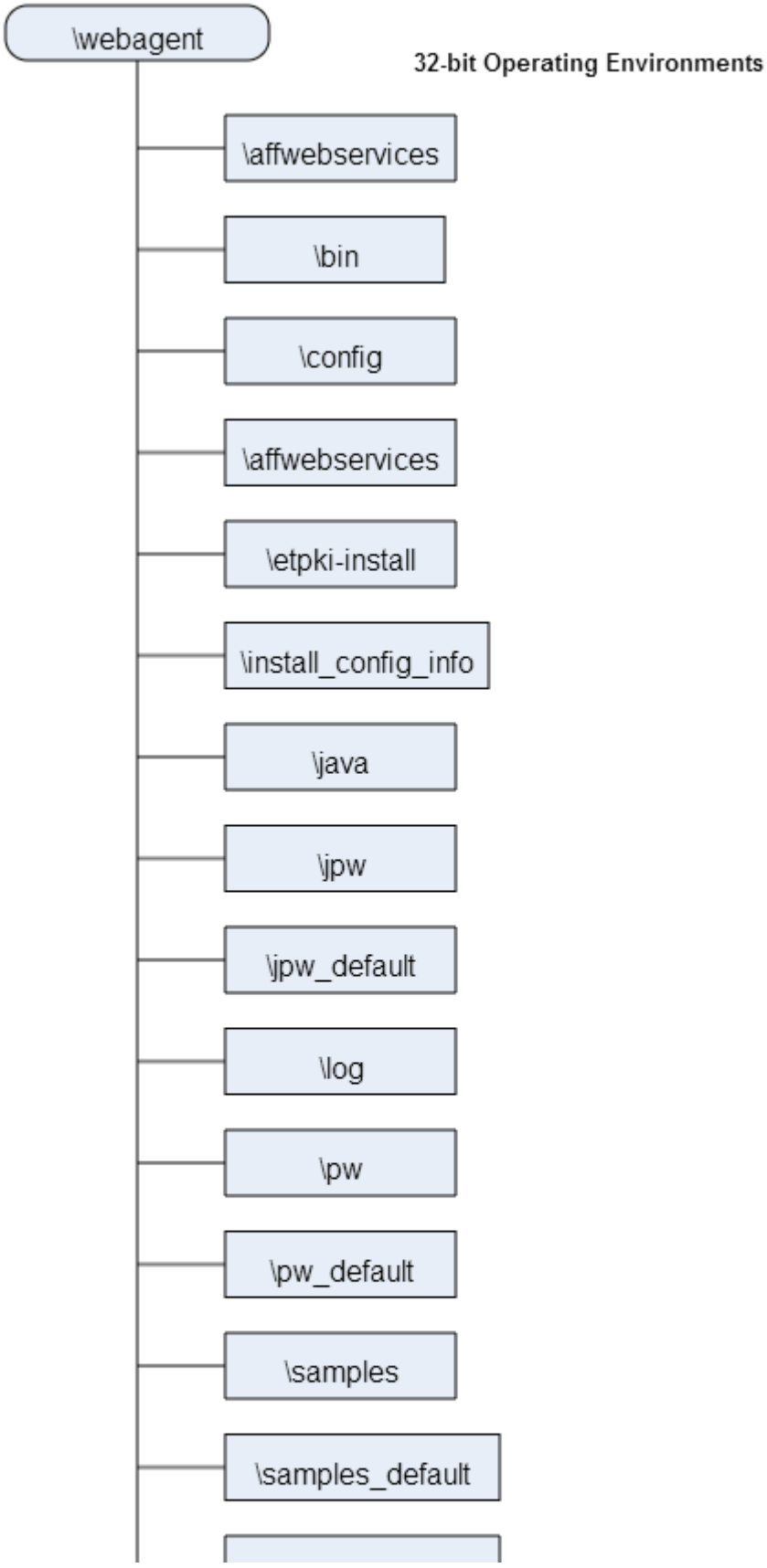
## Hardware Requirements for IIS SiteMinder WSS Agents

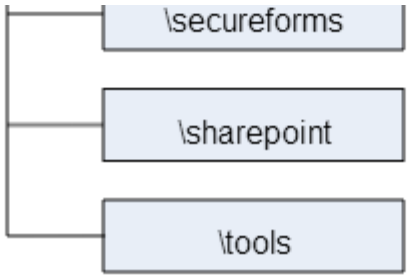
- **Windows**
  CA Single Sign-on WSS Agent operating on Windows require the following hardware:
  - CPU: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in the temporary location.

## Multiple SiteMinder WSS Agent Directories on IIS Servers

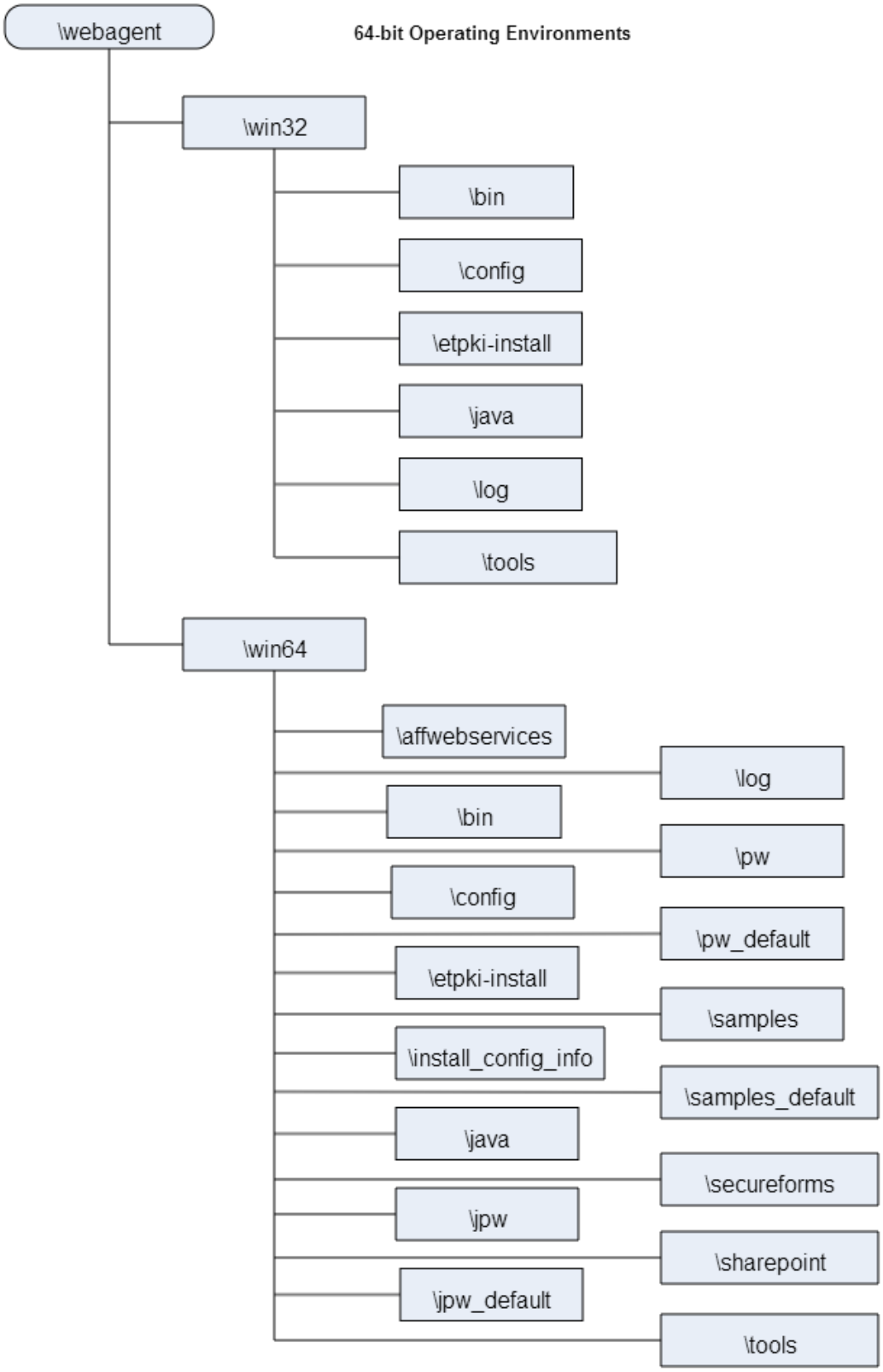
The directory structure added to your IIS web server for your Agent files varies according to the operating environment of your IIS web server. The following directory structures exist:

- CA Single Sign-On Web Agents and CA Single Sign-on WSS Agents for IIS use the directory structure shown in the following illustration:


\webagent
32-bit Operating Environments
\affwebservices
\bin
\config
\affwebservices
\etpki-install
\install_config_info
\java
\jpw
\jpw_default
\log
\pw
\pw_default
\samples
\samples_default




- CA Single Sign-On Agents for IIS installed on 64-bit operating environments use the directory structure shown in the following illustration:


\webagent
64-bit Operating Environments
\win32
\bin
\config
\etpki-install
\java
\log
\tools
\win64
\affwebservices
\log
\bin
\pw
\config
\pw_default
\etpki-install
\samples
\install_config_info
\samples_default
\java
\secureforms
\jpw
\sharepoint
\jpw_default
\tools

# How to Prepare for a WSS Agent for IIS Installation on Your Web Server

**Contents**



To prepare for an WSS Agent for IIS installation on a Windows operating environment, do the following procedures:

## Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.
2. Double-click System.
3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

## Verify that you have an Account with Administrative Privileges on the Windows Computer Hosting your IIS Web Server

To install or configure a CA Single Sign-on WSS Agent on an IIS web server, you need an account with Administrator privileges.

On Windows 2008 systems, do one of the following actions to install or configure a CA Single Sign-on WSS Agent:

- If you are using Windows Explorer, right-click the .exe file. Then select Run as Administrator.
- If you are using a command line, open a new console window with administrative privileges. Then run the command that you want.

> **Note**: For more information about installing or configuring CA Single Sign-on WSS Agents on Windows 2008 systems, see the CA Single Sign-On Web Services Security Release Notes.

## Verify that the IIS Role and Role Services are Installed

The IIS (web server) role is *not* enabled by default. Verify that the IIS role is installed and enabled on each Windows system, before installing the Agent for IIS.

**Follow these steps:**

1. Click Start, All Programs, Administrative Tools, Server Manager.
2. Verify that IIS appears in the Roles list.
3. If the Web Server (IIS) role is not shown, add it using the Add Roles wizard. If you decide to use the ISAPI-filter functions of the Agent for IIS, add the following role services too:
   - CGI
   - ISAPI Extensions
   - ISAPI Filters
   - IIS Management Console
   - Windows Authentication (for the CA Single Sign-On Windows Authentication Scheme)

## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Verify that the Windows IIS Web Server has the Latest Service Packs and Updates

We recommend using Windows Update to verify that your Windows operating environment contains the latest Service Packs and updates, before installing a CA Single Sign-On Agent for IIS.

## Verify that the Microsoft Visual C++ 2005 Redistributable Package (x64) is Installed

Before installing a CA Single Sign-On Agent on a Windows 64-bit platform, download and install the Microsoft Visual C++ 2005 Redistributable Package (x64). Go to the Microsoft downloads page ( http://www.microsoft.com/downloads/Search.aspx), and then search for "Microsoft Visual C++ 2005 Redistributable Package (x64)."

## Review the Policy Server Prerequisites for Agent for IIS Installations

Your Agent for IIS needs the following information about the Policy Servers to which it connects:

- The IP addresses of the Policy Servers
- Certain CA Single Sign-On object names in the Policy Server

The Administrative UI creates these objects in the Policy Server. We recommend creating them before installing your agent to avoid going between your web server and the Administrative UI interfaces later.

Agents for IIS require the names of the following CA Single Sign-On objects stored the Policy Server:

- **Host Configuration Object**
  Contains the settings that the agent uses for subsequent connections to a Policy Server following the initial connection that the agent made.
- **Admin User Name**
  Identifies the name of a CA Single Sign-On user with the following privileges:
  - Administrative privileges
  - Trusted host registration privileges
- **Admin Password**
  Identifies a password that is associated with the Admin User Name in the CA Single Sign-On Policy Server.
- **AgentName**
  Defines the identity of the Web Agent. This identity establishes a mapping between the name and the IP address of each web server instance hosting an Agent.
  When no matching value exists, the agent uses the value of from the DefaultAgentName parameter instead.

**Note**: This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add the parameter name and a value to separate lines in the file.
**Default**: No default
**Limit**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
**Limits**: Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. Not case-sensitive. For example, the names MyAgent and myagent are treated the same.
**Example**: myagent1,192.168.0.0 (IPV4)
**Example:** myagent2, 2001:DB8::/32 (IPV6)
**Example**: myagent, www.example.com (http://www.example.com)

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.

2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider. IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.

3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

# How to Install and Configure WSS Agents for IIS Servers

Installing and configuring the WSS Agent for IIS involves several separate procedures. To install and configure the agent for IIS, use the following process:

1. If you are deploying the Agent for IIS to an IIS server farm, review the following topics:
   - IIS 7.x web server shared configuration (see page 699).
   - How web agent logs and trace logs work with shared configuration (see page ).
2. Gather the information for the installation program (see page ).
3. Gather the information for the configuration program (see page ).
4. Run the Web Services Security installation program (see page ).
5. Run the wizard based configuration program (see page 707).
6. (Optional) Install and configure additional Agents for IIS silently (see page 708).
7. (Optional) Add (see page ) or remove (see page ) Web Services Security protection from virtual sites on IIS web servers silently.
8. Determine if your Agent for IIS requires any manual configuration steps (see page 715).

## IIS 7.x Web Server Shared Configuration and the Agent for IIS

IIS 7.x web servers support shared configurations that streamline the configuration process for an IIS a server farm.

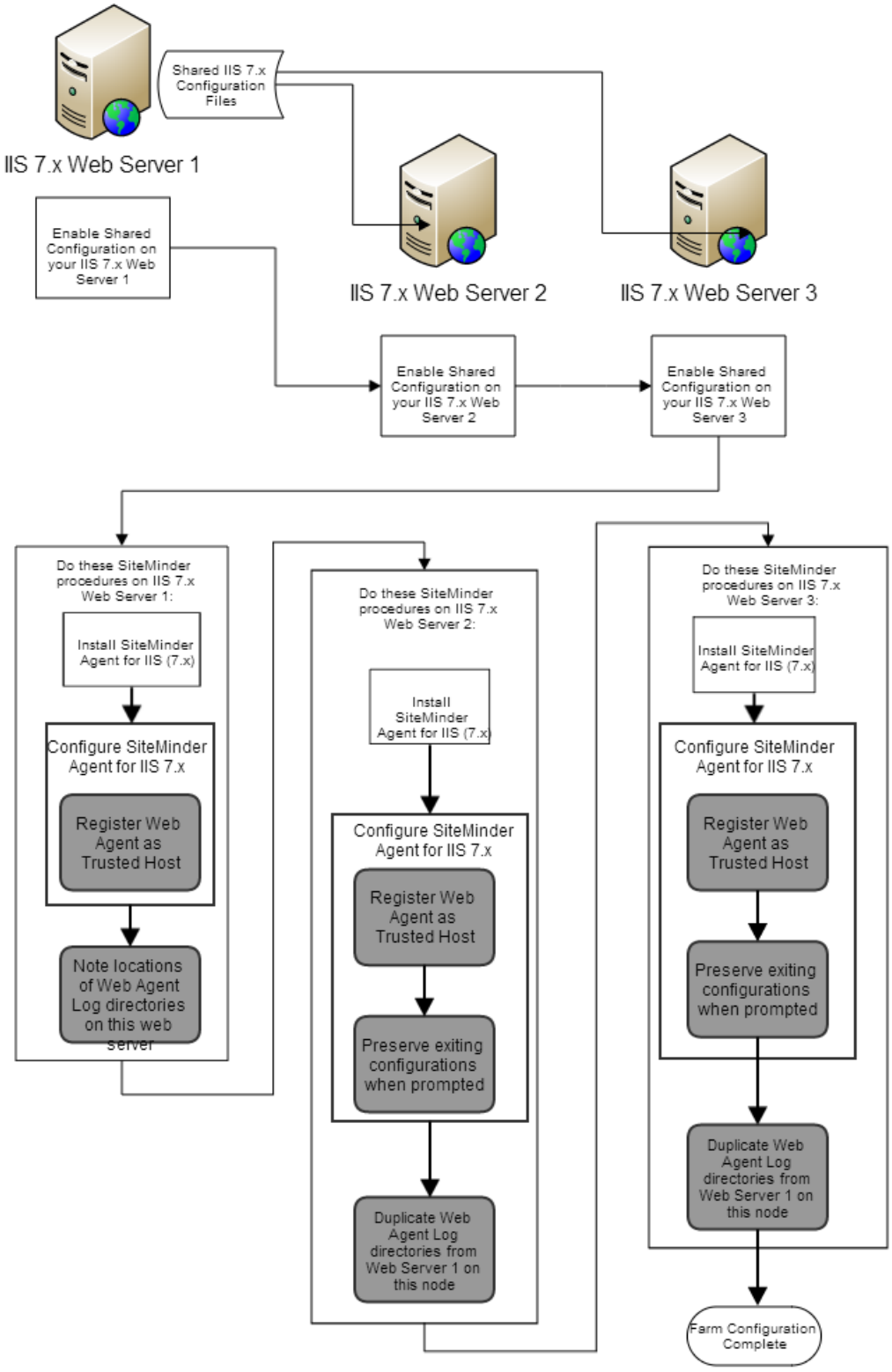
The Agent for IIS can protect resources on IIS server farms that use the shared configuration feature of IIS 7.x.

**Note**: This feature works *only* with the WSS Agent for IIS 7. Older versions of the Agent do *not* support this feature.

IIS 7.x uses network shares to propagate the configuration information across the server farm. The Agent for IIS, however, *cannot* operate on network shares. Using an Agent for IIS on an IIS server farm involves several separate procedures.

For example, suppose you have three IIS 7.x web servers, with all of them using a shared configuration. Web server number one is your primary web server, which contains the configuration information for the farm. Web servers 2 and 3 are nodes that connect to the network share on web server one to read the configuration information.

The entire installation and configuration process for using the WSS Agent for IIS on all three IIS 7.x web servers is described in the following illustration:


Shared IIS 7.x Configuration Files
IIS 7.x Web Server 1
Enable Shared Configuration on your IIS 7.x Web Server 1
IIS 7.x Web Server 2
IIS 7.x Web Server 3
Enable Shared Configuration on your IIS 7.x Web Server 2
Enable Shared Configuration on your IIS 7.x Web Server 3
Do these SiteMinder procedures on IIS 7.x Web Server 1:
Install SiteMinder Agent for IIS (7.x)
Configure SiteMinder Agent for IIS 7.x
Register Web Agent as Trusted Host
Note locations of Web Agent Log directories on this web server
Do these SiteMinder procedures on IIS 7.x Web Server 2:
Install SiteMinder Agent for IIS (7.x)
Configure SiteMinder Agent for IIS 7.x
Register Web Agent as Trusted Host
Preserve exiting configurations when prompted
Duplicate Web Agent Log directories from Web Server 1 on this node
Do these SiteMinder procedures on IIS 7.x Web Server 3:
Install SiteMinder Agent for IIS (7.x)
Configure SiteMinder Agent for IIS 7.x
Register Web Agent as Trusted Host
Preserve exiting configurations when prompted
Duplicate Web Agent Log directories from Web Server 1 on this node
Farm Configuration Complete

## How WSS Agent Logs and Trace Logs Work with IIS 7.x Web Server Shared Configuration

For WSS Agents for IIS running on an IIS server farm, create duplicate log and trace file directories on each node if all the following conditions are true:

- Your Agent for IIS log and trace log directories are specified in an Agent Configuration Object on the Policy Server (*not* in a local configuration file).
- Any of the WSS Agents for IIS in your IIS 7.x web servers in the server farm share the same Agent Configuration object
- Your Agent for IIS log file and trace log directories specified in the shared Agent Configuration Object are *different* than the default setting:

  `agent_home\log`

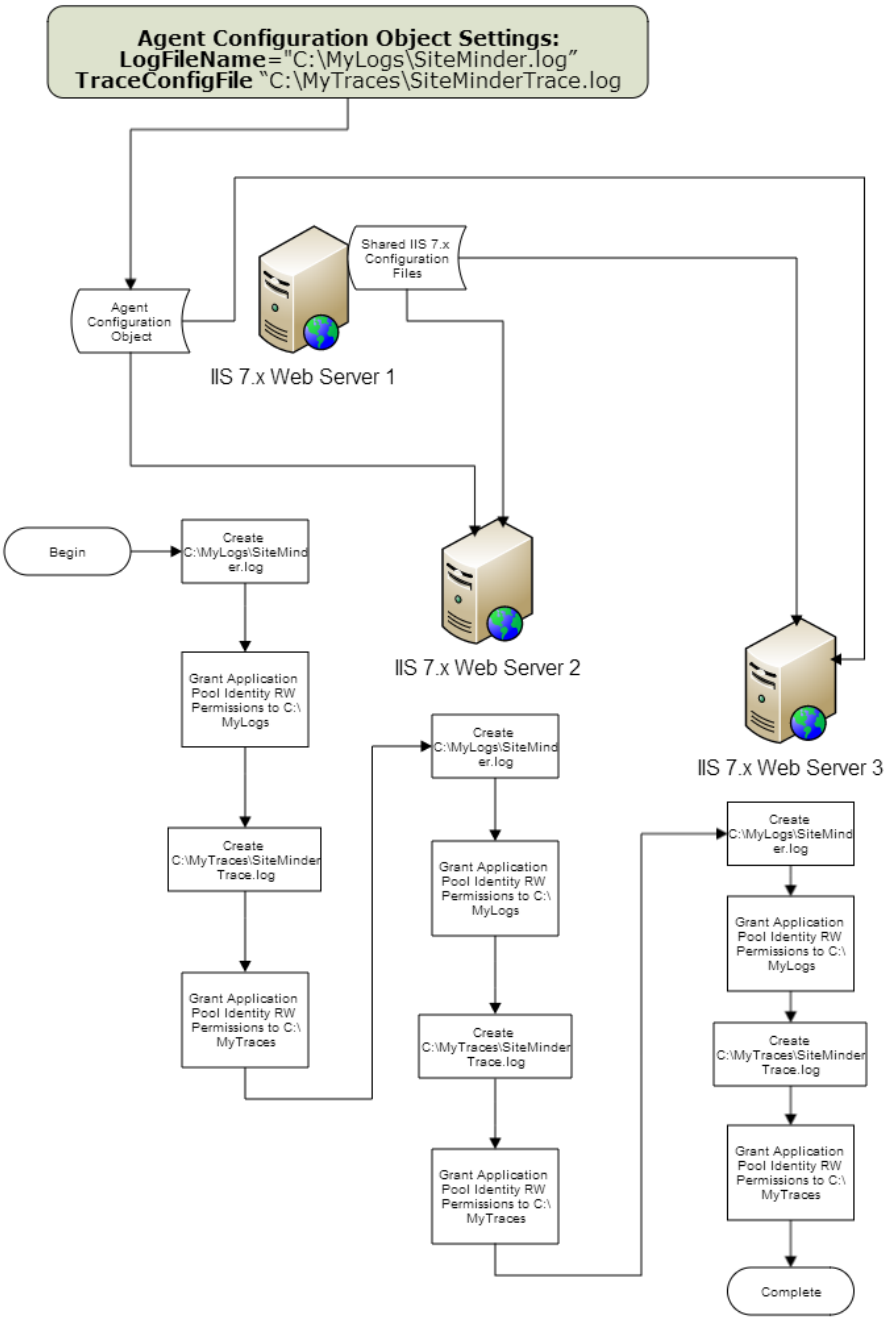
  - **agent_home**
    Indicates the directory where the WSS Agent is installed on your web server.
    **Default** (Windows 32-bit WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
    **Default** (Windows 64-bit WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
    **Default** (Windows 32-bit WSS Agent installations operating on 64-bit systems: C:\Program Files (x86)\CA\Web Services Security\webagent\win32

If all of the previous conditions exist in your server farm, use the following process to enable your WSS Agent logs and trace logs:

1. Create a custom log directory on the IIS 7.x web server that contains the shared configuration for the farm.
2. Grant the application pool identities associated with your protected resources the following permissions to the custom directory on the previous IIS 7.x web server.
   - Read
   - Write
3. Create the same custom log directory on a IIS 7.x web server node in the farm.
4. Grant the application pool identities associated with your protected resources the following permissions to the custom directory on the a IIS 7.x web server node in the farm.
   - Read
   - Write
5. Repeat steps 3 and 4 on all other nodes in your server farm.

For example, suppose you have three IIS 7.x web servers, with all of them using a shared configuration. Web server number one is your primary web server, which contains the configuration information for the farm. Web servers 2 and 3 are nodes that connect to the network share on web server one to read the configuration information.

The entire process for configuring these logs is described in the following illustration:



## Gather Information for the WSS Agent Installation Program

Before running the installation program for the Agent for IIS on the Windows operating environment, gather the following information about your web server:

- **Installation Directory**
  Specifies the location of the WSS Agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: CA Single Sign-On requires the name "webagent" for the bottom directory in the path.

- **Shortcut Location**
  Specifies the location in your Start menu for the shortcut for the Web Agent Configuration wizard.

## Gather the Information for the WSS Agent Configuration Program for IIS Web Servers

Before configuring a WSS Agent on an IIS web server, gather the following information about your CA Single Sign-On environment.

Host Registration

Indicates whether you want to register this agent as a trusted host with a Policy Server. Only one registration per agent is necessary. If you are installing the WSS Agent for IIS 7.x on an IIS server farm, register all IIS agents in the farm as trusted hosts.

**Limits**: Yes, No

- **Admin User Name**
  Specifies the name of a CA Single Sign-On user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Admin Password**
  Specifies the password that is associated with the CA Single Sign-On user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Confirm Admin Password**
  Confirms the password that is associated with the CA Single Sign-On user account that has sufficient privileges to create and register trusted host objects on the Policy Server.

- **Enable Shared Secret Rollover**
  Indicates whether the Policy Server generates a new shared secret when the agent is registered as a trusted host.

- **Trusted Host Name**
  Specifies a unique name for the host you are registering. After registration, this name appears in the list of Trusted Hosts in the Administrative UI. When configuring a WSS Agent for IIS on an IIS web server farm, specify a *unique* name for *each* IIS server node on the farm. For example, if your farm uses six servers, specify six unique names.

- **Host Configuration Object**
  Indicates the name of the Host Configuration Object that exists on the Policy Server.

- **IP Address**
  Specifies the IP addresses of any Policy Servers to which the agent connects. Add a port number if you are *not* using the default port for the authentication server. Non-default ports are used for all three Policy Server connections (authentication, authorization, accounting).
  **Default**: (authentication port) 44442
  **Example:** (IPv4) 127.0.0.1,55555
  **Example:** (IPv6) [2001:DB8::/32][:55555]

  **Note:** If a hardware load balancer is configured to expose Policy Servers in your environment through a single Virtual IP Address (VIP), enter the VIP.

- **FIPS Mode Setting**
  Specifies *one* of the following algorithms:
  FIPS Compatibility/AES Compatibility
  Uses algorithms existing in previous versions of CA Single Sign-On to encrypt sensitive data and is compatible with previous versions of CA Single Sign-On. If your organization does not require the use of FIPS-compliant algorithms, use this option.
  FIPS Migration/AES Migration
  Allows a transition from FIPS-compatibility mode to FIPS-only mode. In FIPS-migration mode, CA Single Sign-On environment continues to use existing CA Single Sign-On encryption algorithms as you reencrypt existing sensitive data using FIPS-compliant algorithms.
  FIPS Only/AES Only
  Uses only FIPS-compliant algorithms to encrypt sensitive data in the CA Single Sign-On environment. This setting does not interoperate with, nor is backwards-compatible with, previous versions of CA Single Sign-On.
  **Default**: FIPS Compatibility/AES Compatibility

Note: FIPS is a US government computer security standard that accredits cryptographic modules which meet the Advanced Encryption Standard (AES).

Important! Use a compatible FIPS/AES mode (or a combination of compatible modes) for both the WSS agent and the Policy Server.

- **Name**
  Specifies the name of the SmHost.conf file which contains the settings the Web Agent uses to make initial connections to a Policy Server.
  **Default**: SmHost.conf

- **Location**
  Specifies the directory where the SmHost.conf file is stored. On Windows 64-bit operating environments, the configuration program creates two separate files. One file supports 64-bit applications, and the other file supports 32-bit applications running on the same web server.
  **Default**: (Windows IIS 7.x 32-bit) *agent_home*\win32\bin\IIS
  **Default**: (Windows IIS 7.x 64-bit) *agent_home*\win64\bin\IIS

- **Virtual Sites**
  Lists the web sites on the IIS 7.x web server that you can protect with CA Single Sign-On.

- **Overwrite, Preserve, Unconfigure**
  Appears when the WSS Agent configuration wizard detects *one* of the following situations:

  - IIS 7.x websites that CA Single Sign-On already protects on a stand-alone IIS web server.
  - IIS 7.x websites that CA Single Sign-On protects on an IIS server farm using shared configuration.

  Select *one* of the following options:
  Overwrite
  Replaces the previous configuration of the WSS Agent with the current configuration.
  Preserve
  Keeps the existing configuration of your WSS Agent. No changes are made to this web server instance. Select this setting for each web server node if you are configuring the WSS Agent for IIS 7.x on an IIS server farm.
  Unconfigure
  Removes the existing configuration of a WSS Agent from the web server. Any resources are left unprotected by CA Single Sign-On.
  **Default**: Preserve

**Important!** Do not configure and unconfigure virtual sites at the same time. Run the wizard once to configure the sites you want, and then run the wizard again to unconfigure the sites you want.

- **Agent Configuration Object Name**
  Specifies the name of an Agent Configuration Object (ACO) already defined on the Policy Server. IIS web servers in a server farm using shared configuration support sharing a single ACO name with all IIS servers in the farm.
  **Default**: AgentObj

## Run the Installer to Install a WSS Agent

Install the WSS Agent using the Web Services Security installation media on the Technical Support site.

**Follow these steps:**

1. Exit all applications that are running.

2. Navigate to the installation material.

3. Double-click ca-sm-wss-version-*cr*-win32.exe.

   - **cr**
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

   The Web Services Security installation wizard starts.

   > **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

4. Use gathered system and component information to install the WSS Agent. Consider the following points when running the installer:

   - When prompted to select which Web Services Security Agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - If the installer detects the presence of an existing CA Single Sign-On Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

5. Review the information that is presented on the Pre-Installation Summary page, then click Install.

   > **Note:** If the installation program detects that newer versions of certain system DLLs are installed on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location.

6. On the CA Single Sign-On Web Services Security Configuration screen, click one of the following options and click Next:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.

- No. I will configure CA Single Sign-On Web Services Security Agents later.

If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

**Important!** For WSS Agents for Web Servers installed on IIS servers, reboot your system after installation; it is not sufficient to restart the IIS service. Also, do not configure the Agent immediately after installation; there are some tasks you must do before configuring the Agent.

7. Click Done.
   If you selected the option to configure WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process. Use the information that you gathered earlier to complete the wizard.
   If you did not select the option to configure WSS Agents now, or if you are required to reboot the system after installation, run the configuration wizard manually later.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log

  - **WSS_Home**
    Specifies the path to where Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security

  - **install-date-and-time**
    Specifies the date and time that the WSS Agent was installed.

- The Agent cannotcommunicate properly with the Policy Server until the trusted host is registered.

## Run the WSS Agent Configuration Wizard

After gathering the information for your Agent Configuration worksheet, run the Agent Configuration wizard. The configuration wizard creates a runtime instance of the agent for IIS on your IIS web server.

Running the configuration wizard once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment and settings.

**Note**: The configuration wizard for this version of the Agent for IIS does *not* support console mode.

**Follow these steps:**

1. Open the following directory on your web server:

   *WSS_Home*\install_config_info

   - ***WSS_Home***
     Specifies the path to where Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Right-click ca-pep-config.exe, and then select Run as administrator.

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

   The WSS Agent Configuration wizard starts.

3. Use the information you gathered earlier to complete the wizard.

# Run the Unattended or Silent Installation and Configuration Programs for your Agent for IIS

**Contents**



The unattended or silent installation option can help you automate the installation and configuration process. This method saves time if you have a large CA Single Sign-On Web Services Security environment that uses many agents with identical settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

**Follow these steps:**

1. Run the following wizards on your first IIS web server (in the order shown):

   a. The CA Single Sign-On Web Services Security Installation wizard.

   b. The CA Single Sign-On Web Services Security Configuration wizard.

2. Locate the following file on your first IIS web server:

   ```
   WSS_home\install_config_info\ca-wss-installer.properties
   ```

   **Note**: If the path contains spaces, surround it with quotes.

   - **WSS_Home**
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

3. Perform each of the following steps on the other IIS web server nodes in your environment:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

   a. Create a temporary directory on an IIS web server node.

   b. Copy the following files from your first IIS web server (from Steps 1 and 2) to the temporary directory on your other IIS web server:

      - The CA Single Sign-on WSS Agent Installation executable file.
      - The ca-pepconfig-installer.properties file.

   c. Open a Command Prompt window with Administrative privileges in the temporary directory.

   d. Run the following command:

      ```
      agent_executable -f properties_file -i silent
      ```

      The CA Single Sign-on WSS Agent for IIS is installed and configured on the node automatically.

   e. (Optional) Delete the temporary directory from your web server node.

4. Repeat Step 3 for each additional web server in your CA Single Sign-On environment that uses the configuration that the settings in your ca-wss-installer.properties file specify.

## Add CA Single Sign-On Web Services Security Protection to Additional Virtual Sites on IIS Web Servers Silently

If your IIS web server already has a CA Single Sign-on WSS Agent for IIS installed, you can protect any additional virtual websites on the web server. For example, if you add two new virtual sites named Example2 and Example3 to your IIS server, you can protect web services on them with CA Single Sign-On Web Services Security.

If you do not want to run configuration wizard, or if you have many IIS web servers in a server farm, use the silent mode.

The CA Single Sign-On Web Services Security configuration program supports a silent or unattended mode that requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   ```
   WSS_Home\install_config_info\ca-wss-installer.properties
   ```

   - **WSS_Home**
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Perform each of the following steps on the IIS web servers to which you want to protect the additional virtual sites:

   **Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.**Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

   Create a temporary directory on an IIS web server node.

   Copy the following files from your first IIS web server (from Steps 1 and 2) to the temporary directory on your IIS web server node:

   CA Single Sign-on WSS Agent configuration executable file (ca-pep-config.exe).

   CA Single Sign-on WSS Agent ca-wss-installer.properties file.

   Open the ca-wss-installer.properties file with a text editor.

   Locate the following parameter:

   CONFIGURE_SITES=

   Specifies the names of IIS 7.x web sites to protect on an IIS 7.x web server. Verify that these names match those names shown in under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.

   For more information, see the comments in the ca-wss-installer.properties file.

   **Example**: Default Web Site,Example1,Example2

   Add the names of the web sites you want to configure to the previous parameter. Remove the names of any other sites on the web server that you want to leave unchanged.

Locate the following parameter:

HOST_REGISTRATION_YES=

Specifies if the agent configuration program registers the agent with a Policy Server. Each web server requires only one trusted host registration is required. Set the value of this parameter to 0 if you have previously registered a web server with the Policy Server as a trusted host.

**Default**: 1 (yes)

**Limits**: 0 (no registration), 1 (registration)

If the IIS web *server* is *already* registered as a trusted host with the CA Single Sign-On Policy Server, change the value of the previous parameter to 0. Otherwise, the configuration program registers the web server as a trusted host.

Open a Command Prompt window with Administrative privileges in the temporary directory.

**Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

Run the following command:

ca-pep-config.exe -f ca-wss-installer.properties -i silent

The CA Single Sign-on WSS Agent for IIS is installed and configured on the node automatically.

(Optional) Delete the temporary directory from your web server node.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the configuration specified by the settings in your ca-wss-installer.properties file.

## Remove a CA Single Sign-on WSS Agent Configuration from an IIS Web Server Silently

To remove the CA Single Sign-On Web Services Security protection from all the websites on an IIS web server without using the CA Single Sign-On Web Services Security configuration wizard, use silent or unattended mode. This mode requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   `WSS_Home\install_config_info\ca-wa-installer.properties`

   - **WSS_Home**
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Perform each of the following steps on the IIS web servers to which you want to remove protection from virtual sites:

**Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.**Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

Open the following directory on an IIS web server node.

*WSS_Home*\install_config_info

Copy the ca-wss-installer.properties file from your first IIS web server (from Step 1) to the install_config_info directory on your IIS web server node.

Open the ca-wss-installer.properties file with a text editor.

Locate the following parameter:

UNCONFIGURE_SITES=

Specifies the names of IIS 7.x web sites from which to remove CA Single Sign-On Web Services Security protection on an IIS 7.x web server. Verify that these names match those names shown in under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.

Removing the CA Single Sign-on WSS Agent configuration from a website leaves its resources *unprotected*.

For more information, see the comments in the ca-soasm-installer.properties file.

**Example**: Default Web Site,Example4,Example5

Enter the names of the websites you want to unconfigure in the previous parameter.

Locate the following parameter:

CONFIGURE_SITES=

Specifies the names of IIS 7.x web sites to protect on an IIS 7.x web server. Verify that these names match those names shown in under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.

For more information, see the comments in the ca-wss-installer.properties file.

**Example**: Default Web Site,Example1,Example2

Verify that the previous parameter contains no website names.

Open a command prompt window with Administrative privileges.

**Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

Run the following command:

ca-pep-config.exe -f *properties_file* -i silent

The websites are unconfigured on the node automatically.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the configuration specified by the settings in your ca-wss-installer.properties file.

## Remove CA Single Sign-On Web Services Security Protection From Some Virtual Sites on IIS Web Servers Silently

If your IIS web server already has a CA Single Sign-on WSS Agent for IIS installed, you can remove protection from some virtual websites on the web server. For example, suppose you want to remove protection from only two of the virtual sites named Example4 and Example5 from to your IIS server. Modify the ca-wss-installer.properties file to remove the configuration from those two virtual websites while leaving the protection for the other websites unchanged.

If you do not want to run the configuration wizard, or if you have many IIS web servers in a server farm, use the silent mode.

The CA Single Sign-On Web Services Security configuration program supports a silent or unattended mode that requires no interaction from the end user.

**Follow these steps:**

1. Locate the following file on your first IIS web server.

   `WSS_Home\install_config_info\ca-wa-installer.properties`

   - **WSS_Home**
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Perform each of the following steps on the IIS web servers from which you want to remove the protection of the additional virtual sites:

**Note**: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.**Note**: In this context, the first server refers to the IIS web server in a farm where the shared configuration information is stored. A node refers to any other IIS web servers in the farm which read the shared configuration from the first server.

Copy the ca-wss-installer.properties file from your first IIS web server (from Step 1) to the install_config_info directory on your IIS web server node:

Open the ca-wss-installer.properties file with a text editor.

Locate the following parameter:

UNCONFIGURE_SITES=

Specifies the names of IIS 7.x web sites from which to remove CA Single Sign-On Web Services Security protection on an IIS 7.x web server. Verify that these names match those names shown in under the Sites folder in the Internet Information Services (IIS) Manager of your web server. Separate multiple website names with commas.

Removing the CA Single Sign-on WSS Agent configuration from a website leaves its resources *unprotected*.

For more information, see the comments in the ca-soasm-installer.properties file.

**Example**: Default Web Site,Example4,Example5

Add the names of the web sites from which you want to remove the configuration to the previous parameter. Remove the names of any other sites on the web server that you want to leave unchanged.

Locate the following parameter:

HOST_REGISTRATION_YES=

Specifies if the agent configuration program registers the agent with a Policy Server. Each web server requires only one trusted host registration is required. Set the value of this parameter to 0 if you have previously registered a web server with the Policy Server as a trusted host.

**Default**: 1 (yes)

**Limits**: 0 (no registration), 1 (registration)

If the IIS web *server* is *already* registered as a trusted host with the CA Single Sign-On Policy Server, set the previous parameter to 0. Otherwise, the configuration program registers the web server as a trusted host.

Open a Command Prompt window with Administrative privileges in the temporary directory.

**Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

Run the following command:

ca-pep-config.exe -f ca-wss-installer.properties -i silent

The CA Single Sign-On Web Services Security configuration is removed from the selected virtual sites on the node automatically.

3. Repeat Step 2 for each additional IIS web server node in your environment that uses the configuration specified by the settings in your ca-wss-installer.properties file.

# How to Configure Certain Settings for the SiteMinder WSS Agent for IIS Manually

**Contents**



In some situations, the WSS Agent configuration programs *cannot* add the proper settings to all the IIS web server directories which need them.

Configure the WSS Agent for IIS settings manually in *any* of the following situations:

- Your Agent for IIS log files are not stored in the following default directory (see page 716):

  `WSS_agent_home\log`

  - **agent_home**
    Indicates the directory where the WSS Agent is installed on your web server.
    **Default** (Windows 32-bit WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
    **Default** (Windows 64-bit WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
    **Default** (Windows 32-bit WSS Agent installations operating on 64-bit systems: C:\Program Files (x86)\CA\Web Services Security\webagent\win32

  For example, suppose that you store your log files in the C:\My Logs\SiteMinder directory. Grant this directory permissions.

- You use a Web Services Security authentication scheme which requests or requires client certificates (see page ).

## Set Permissions Manually for Non-Default Log Locations

If you decide to store your agent log files in a non default directory, grant your application pools permissions to the directory. For example, if you want to store your log files in a directory named C:\MyLogFiles, grant permissions for all your application pool identities to C:\MyLogFiles.

Microsoft provides a command line utility, icacls.exe you can use to set the appropriate permissions. This procedure provides one possible example of a way to set permissions using tools or utilities provided by third-party vendors.

**Important!** CA provides this information only as an example of one possible method of configuring CA Single Sign-On without using the programs and utilities tested and approved by CA. Microsoft provides the icacls.exe command as part of the Windows operating environment. You may choose to use the following examples as a guide to grant file permissions for the agent for IIS. This command and the syntax shown are subject to change by Microsoft at any time and without notice. For more information, go to the Microsoft Support (http://support.microsoft.com/) website, and search for "icacls"

**Follow these steps:**

1. Open a Command Prompt Window on your IIS web server.

   **Important!** Before running a CA Single Sign-On utility or executable on Windows Server 2008, open the command-line window with administrator permissions. Open the command-line window this way, even if your account has administrator privileges.

2. Run the icacls command. Use the following example as a guide:

   ```
   icacls log_directory /grant IIS AppPool\application_pool_identity
   ```

   - **log_directory**
     Specifies the non default log directory to which you must grant permissions.

   - **application_pool_identity**
     Specifies the identity of the application pool associated with the application protected by CA Single Sign-On on your IIS web server.

3. Repeat Step 2 for each application pool identity on your IIS web server. For example, if you have two application pools, grant permissions to both.

4. If you have an IIS server farm using Shared Configuration, repeat Steps 1 through 3 for each IIS web server in the farm.
   The permissions are set.

## Change IIS Settings Manually for CA Single Sign-On Web Services Security Authentication Schemes Requiring Certificates

If you use CA Single Sign-On Web Services Security authentication schemes that request or require certificates, change the settings for the following virtual directories:

- cert
- certoptional

**Follow these steps:**

1. Open IIS manager.
2. Expand your web server.
3. The Application pools icon and Sites folder appear.
4. Expand Sites.
   A list of web sites appears.
5. Expand the website that is associated with your authentication scheme that requires certificates.
   The siteminderagent virtual folder appears.
6. Expand the siteminderagent virtual folder.
   A list of subfolders appears.
7. Click the cert folder.
   The settings icons appear.
8. Double-click SSL Settings.
   The SSL Settings page appears.
9. Select the Require SSL check box, and then click the Require option button.
10. Under Actions, click Apply.
    The changes are applied.
11. Click the certoptional folder.
    The settings icons appear.
12. Double-click SSL Settings.
    The SSL Settings page appears.
13. Click the Accept option button.
14. Under Actions, click Apply.
    The changes are applied.
15. Repeat Steps 3 through 14 for other websites on your IIS web server that require certificates.

16. For IIS server farms using Shared Configuration, repeat Steps 1 through 15 on each IIS web server in your farm.
    The settings are changed.

# Uninstall a SiteMinder WSS Agent from IIS Servers

**Contents**



## Set JRE in PATH Variable Before Uninstalling the CA Single Sign-On Agent

On Windows and UNIX systems, when you are uninstalling a CA Single Sign-On Agent, make sure the JRE is in the PATH variable or the uninstallation program stops and issues one of the following error messages:

- “Could not find a valid Java virtual machine to load. You need to reinstall a supported Java virtual machine.”
- "No Java virtual machine could be found from your PATH environment variable. You must install a VM prior to running this program."

**Follow these steps:**

**On Windows**

1. Go to the Control Panel.
2. Double-click System.
3. In the Environment Variables dialog, add the location of the JRE to the PATH system variable.
   For example, C:\j2sdk*version_number*\jre\bin

**On UNIX**

Run the following commands:

1. PATH=$PATH:*JRE*/bin
   - **JRE**
     Specifies the location of your JRE.
     For example, /usr/bin/j2sdk*version_number*/jre
2. export PATH

## Uninstall a CA Single Sign-on WSS Agent

To uninstall a CA Single Sign-on WSS Agent, run the CA Single Sign-On Web Services Security uninstall wizard.

**Follow these steps:**

1. Navigate to the WSS_*HOME*\install_config_info (Windows) or WSS_*HOME*/install_config_info (UNIX) directory and run the CA Single Sign-On Web Services Security uninstall wizard to remove core CA Single Sign-On Web Services Security components:

   - Windows: wss-uninstall.cmd
   - UNIX: wss-uninstall.sh
   - **WSS_HOME**
     Specifies the CA Single Sign-On Web Services Security installation location.

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

   The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.

3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.
   The uninstall wizard removes all selected CA Single Sign-On Web Services Security components.

4. Restart the server.

# SiteMinder WSS Agent Logging for IIS Servers

**Contents**



## Logs of Start-up Events

To assist in debugging, startup events are recorded in a log. Each message may provide clues about the problem. These logs are stored in the following locations:

- On Windows systems, these events are recorded in the Windows Application Event log.

- On UNIX systems, these events are sent to STDERR. Apache servers map STDERR to the Apache error_log file, so these events are also recorded in that log.

## Error Logs and Trace Logs

You can use the Web Agent logging function to monitor the performance of the Web Agent and its communication with the Policy Server. The logging feature provides accurate and comprehensive information about the operation of CA Single Sign-On processes to analyze performance and troubleshoot issues.

A log is a record of events that occur during program execution. A log consists of a series of log messages, each one describing some event that occurred during program execution. Log messages are written to log files.

**Note**: IIS Agents create log files only after the first user request is submitted. Apache 2.0 Web Agents create log files when the Apache server starts.

The Web Agent uses the following log files:

- **Error log**
  Contains program and operational-level errors. One example is when the Web Agent cannot communicate with Policy Server. The level of detail output in this log cannot be customized. Error logs contain the following types of messages:

  - **Error messages**
    Contain program-level errors, which indicate incorrect or abnormal program behavior, or an inability to function as expected due to some external problem, such as a network failure. There are also operational-level errors. This type of error is a failure that prevents the operation from succeeding, such as opening a file or authenticating a user.

  - **Informational messages**
    Contain messages for the user or administrator that some event has occurred; that is, that a server has started or stopped, or that some action has been taken.

  - **Warning messages**
    Contain warnings for the user or administrator of some condition or event that is unusual or indicative of a potential problem. This does not necessarily mean there is anything wrong.

- **Trace log**
  Contains detailed warning and informational messages, which you can configure. Examples include trace messages and flow state messages. This file also includes data such as header details and cookie variables. Trace logs contain the following messages:

- **Trace messages**
  Provide detailed information about program operation for tracing and/or debugging purposes. Trace messages are ordinarily turned off during normal operation. In contrast to informational, warning, and error messages, trace messages are embedded in the source code and can not easily be localized. Moreover, trace messages may include significant data in addition to the message itself; for example, the name of the current user or realm.

You specify the location of both the error and trace log files when you configure the Web Agent. Use the error and trace logs to help solve any issues that may prevent the Web Agent from operating properly.



**Note:** For Agents on Windows platforms, set the EnableWebAgent parameter to yes to ensure that the Web Agent log gets created. If you leave EnableWebAgent set to no (the default) and set the logging parameters, the Agent log gets created only for Agents on UNIX platforms.

## Parameter Values Shown in Log Files

Web Agents list configuration parameters and their values in the Web Agent error log file, but there are differences between the ways that Traditional and Framework agents do this.

Framework agents record the configuration parameters and their values in the log file exactly as you entered them in the Agent Configuration Object or the local configuration file. All of the parameters, including those which may contain an incorrect value, are recorded in the log file.

Traditional agents process the parameter values before recording them. If the parameter has a proper value, the parameter and its value are recorded in the log file. Parameters with incorrect values are *not* recorded in the log file.

## Set Up and Enable Error Logging

Error logs require the following settings:

- Logging is enabled.
- A location for the log file is specified.

The parameters that enable error logging and determine options such as appending log data are defined in a local configuration file or an Agent Configuration Object at the Policy Server.

Agents that are installed on an IIS or Apache web servers do not support dynamic configuration of log parameters that are set locally in a local configuration file. The changes take effect when the Agent is restarts. However, these log settings can be stored and updated dynamically in an agent configuration object at the Policy Server.

**Note:** IIS Agents create log files only after the first user request is submitted. Apache 2.0 Web Agents create log files when the Apache server starts.

**Follow these steps:**

1. If you do not have a log file already, create a log file and any related directories.
2. Set the value of the LogFile parameter to yes.

   **Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig.conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.

3. Specify the full path to the error file, including the file name, in any of the following parameters:

   - **LogFileName**
     Specifies the full path (including the file name) of the log file.
     **Default:** No
     **Example:** (Windows) *agent_home*\log\WebAgent.log
     **Example:** (UNIX/LInux) /export/iPlanet/servers/https-jsmith/logs/WebAgent.log

   - **LogFileName32**
     Specifies the full path of a log file for a CA Single Sign-on WSS Agent for IIS (on 64-bit Windows operating environments protecting 32-bit applications). The 32-bit applications run in Wow64 mode on the 64-bit Windows operating environment. If logging is enabled but this parameter is not set, the CA Single Sign-on WSS Agent for IIS appends _32 to the log file name.
     **Default**: No
     **Limits**: For Windows 64-bit operating environments only. Specify the file name at the end of the path.
     **Example**: (Windows 64-bit operating environments using Wow64 mode) *agent_home* \log\WebAgent32.log.

4. (Optional) Set the following parameters (in the Agent Configuration Object on the Policy Server or in the local configuration file):

   - **LogAppend**
     Adds new log information to the end of an existing log file. When this parameter is set to no, the entire log file is rewritten each time logging is invoked.
     **Default:** No

- **LogFileSize**
  Specifies the size limit of the log file in megabytes. When the current log file reaches this limit, a new log file is created. The new log file uses one of the following naming conventions:

  - For framework agents, the new log file has a sequence number that is appended to the original name. For example, a log file named myfile.log is renamed to myfile.log.1 when the size limit is reached.

  - For traditional agents, the new log files are named by appending the date and timestamp to the original name. For example, a log file named myfile.log, is renamed to myfile.log.09-18-2003-16-07-07 when the size limit is reached.

  Archive or remove the old files manually.
  **Default:** 0 (no rollover)
  **Example:** 80

- **LogLocalTime**
  Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
  **Default:** Yes
  If you use a local configuration file, your settings resemble the following example:

```
LogFile="yes"
LogFileName="/export/iPlanet/servers/https-myserver/logs/errors.log"
LogAppend="no"
LogFileSize="80"
LogLocalTime="yes"
```

Error logging is enabled.

## Enable Transport Layer Interface (TLI) Logging

When you want to examine the connections between the agent and the Policy Server, enable transport layer interface logging.

**To enable TLI logging**

1. Add the following environment variable to your web server.

   ```
   SM_TLI_LOG_FILE
   ```

2. Specify a directory and log file name for the value of the variable, as shown in the following example:

   ```
   directory_name/log_file_name.log
   ```

3. Verify that your agent is enabled.

4. Restart your web server.
   TLI logging is enabled.

### Limit the Number of Log Files Saved

You can limit the number of log files that an agent keeps. For example, if you want to save disk space on the system that stores your agent logs, you can limit the number of log files using the following parameter:

LogFilesToKeep

Specifies the number of agent log files that are kept. New log files are created in the following situations:

- When the agent starts.
- When the size limit of the log file (specified by the value of the LogFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing logs files which exceed the number that you want to keep. For example, If your system has 500 log files stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 files.

Setting the value of this parameter to zero retains all the log files.

**Default**: 0

**Follow these steps:**

Archive or delete any existing log files from your system.

Set the value of the LogAppend parameter to no.

Change the value of the LogFilesToKeep parameter to the number of log files that you want to keep.

## How to Set Up Trace Logging

To set up trace logging, use the following process:

1. Set up and Enable Trace logging.
2. Determine what you want to record in the trace log by reviewing the following lists:
   - Trace Log Components and Subcomponents
   - Trace Message Data Fields
   - Data Field Filters
3. Duplicate the default Trace Configuration File.
4. Modify the duplicate file to include the items you want to record.
5. Restart the agent.

## Configure Trace Logging

Before you can use trace logging, you must configure it by specifying a name, location, and parameters for the trace log file. These settings control the size and format of the file itself. After trace logging is configured, you determine the content of the trace log file separately. This lets you change the types of information contained in your trace log at any time, without changing the parameters of the trace log file itself.

**Follow these steps:**

1. Locate the WebAgentTrace.conf file on your web server. Duplicate the file.

   **Note**: If you are running the CA Single Sign-On Agent for IIS and protecting 32-bit applications on a 64-bit system (WoW64 mode), create two duplicates. There are separate directories for 32 and 64-bit applications on 64-bit Windows operating environments.

2. Open your Agent Configuration Object or local configuration file.

3. Set the TraceFile parameter to yes.

   **Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig. conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.

4. Specify the full path to the trace log files in following parameters:

   - **TraceFileName**
     Specifies the full path to the trace log file.
     **Default:** No default
     **Limits**: Specify the file name in this parameter.**Example:***agent_home*\log\trace.log

   - **TraceFileName32**
     Specifies the full path to the trace file for the CA Single Sign-On Agent for IIS is running on a 64-bit Windows operating environment and protecting 32-bit applications. Set this parameter if you have a CA Single Sign-On Agent for IIS installed on a 64-bit Windows operating environment and protecting a 32-bit Windows application. The 32-bit applications run in Wow64 mode on the 64-bit Windows operating environment. If trace logging is enabled but this parameter is not set, the CA Single Sign-on WSS Agent for IIS appends _32 to the file name.
     **Default**: No default.
     **Limits**: For Windows 64-bit operating environments only. Specify the trace file name at

the end of the path.
**Example**: (Windows 64-bit operating environments using Wow64 mode) *agent_home* \log\WebAgentTrace32.log.

5. Specify the full path to the duplicate copies of WebAgentTrace.conf file (you created in Step 1) in the following parameters:

   - **TraceConfigFile**
     Specifies the location of the WebAgentTrace.conf configuration file that determines which components and events to monitor.
     **Default:** No default
     **Example:***agent_home*\config\WebAgentTrace.conf

   - **TraceConfigFile32**
     Specifies the location of the WebAgentTrace.conf configuration file that determines which components and events to monitor. Set this parameter if you have a CA Single Sign-on WSS Agent for IIS installed on a 64-bit Windows operating environment and protecting a 32-bit Windows application. The 32-bit applications run in Wow64 mode on the 64-bit Windows operating environment. If logging is enabled but this parameter is not set, the CA Single Sign-on WSS Agent appends _32 to the file name.
     **Default**: No default.
     **Limits**: For Windows 64-bit operating environments only. Specify the configuration file name at the end of the path.
     **Example**: (Windows 64-bit operating environments using Wow64 mode) *agent_home* \config\WebAgentTrace32.conf.
     **Note:** This file is not used until the web server is restarted.

6. Define the format of the information in your trace log file by setting the following parameters in your Agent Configuration Object or local configuration file:

   - **TraceAppend**
     Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
     **Default:** No

   - **TraceFormat**
     Specifies how the trace file displays the messages. Choose *one* of the following options:

     - default—uses square brackets [] to enclose the fields.
     - fixed—uses fields with a fixed width.
     - delim—uses a character of your choice to delimit the fields.
     - xml—uses XML-like tags. A DTD or style sheet is *not* provided with the Web Agent.

     **Default:** default (square brackets)

   - **TraceDelimiter**
     Specifies a custom character that separates the fields in the trace file.
     **Default:** No default
     **Example:** |

- **TraceFileSize**
  Specifies (in megabytes) the maximum size of a trace file. The Web Agent creates a new file when this limit is reached.
  **Default:** 0 (a new log file is not created)
  **Example:** 20 (MB)

- **LogLocalTime**
  Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
  **Default:** Yes

7. Edit the WebAgentTrace.conf file to include a "components:" entry with value "XMLAgent.". For example:

   ```
   # For WSS Agent
   components: XMLAgent
   data: Date, Time, Pid, Tid, TransactionID, Function, Message.
   ```

   Framework agents do not support dynamic configuration of log parameters set locally in the Agent configuration file. Consequently, when you modify a parameter, the change does not take effect until you restart the web server. However, these log settings can be stored and updated dynamically if you configure them in an Agent configuration object on the Policy Server.

   **Note:** IIS Agents create log files only after the first user request is submitted. Apache 2.0 Web Agents create log files when the Apache server starts.

8. Restart the web server so the CA Single Sign-on WSS Agent uses the new trace configuration file.

## Trace Log Components and Subcomponents

The CA Single Sign-On Agent can monitor specific CA Single Sign-On components. When you monitor a component, all of the events for that component are recorded in the trace log. Each component has one or more subcomponents that the agent can also monitor. If you do not want the agent to record all of the events for a component, you can specify only those subcomponents you want to monitor instead.

For example, if you want to record only the single sign-on messages for an agent on a web server, you would specify the WebAgent component and the SSO subcomponent.

The following components and subcomponents are available:

- **AgentFramework**
  Records all Agent framework messages. (Applies only to framework agents.) The following subcomponents are available:

  - Administration

  - Filter

- HighLevelAgent
- LowLevelAgent
- LowLevelAgentWP

- **AffiliateAgent**
  Records web Agent messages related to the 4.x Affiliate Agent, which is part of Federation Security Services, a separately-purchased product. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **SAMLAgent**
  Web Agent messages related to the SAML Affiliate Agent. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **WebAgent**
  Records all Web Agent log messages. Applies to all Agents *except* IIS 6.0 or Apache 2.0 Agents. The following subcomponents are available:
  - AgentCore
  - Cache
  - authentication
  - Responses
  - Management
  - SSO
  - Filter
- **Agent_Functions**
  Records all Agent API messages. The following subcomponents are available:
  - Init
  - UnInit
  - IsProtected
  - Login
  - ChangePassword
  - Validate
  - Logout

- Authorize
- Audit
- FreeAttributes
- UpdateAttributes
- GetSessionVariables
- SetSessionVariables
- DeleteSessionVariables
- Tunnel
- GetConfig
- DoManagement

- **Agent_Con_Manager**
  Records messages related to internal processing of the Agent API. The following subcomponents are available:
  - RequestHandler
  - Cluster
  - Server
  - WaitQueue
  - Management
  - Statistics

## Trace Message Data Fields

You can define what each trace message for a specific component contains by specifying which data fields to include in the message.

Data fields use the following syntax:

```
data:data_field1,data_field2,data_field3
```

Some data fields are shown in the following example:

```
data:message,date,time,user,agentname,IPAddr
```

There may not be data for fields in each message, so blank fields my occur. For example, if you select RealmOID as a data field, some trace messages will display the realm's OID while others will not.

The following data fields are available:

- **Message**
  Includes the actual trace message
- **SrcFile**
  Includes the source file and line number of the trace message
- **Pid**
  Includes the process ID
- **Tid**
  Includes the thread ID
- **Date**
  Includes the date
- **Time**
  Includes the time
- **PreciseTime**
  Includes the time, including milliseconds
- **Function**
  Includes the function in the code containing the trace message
- **User**
  Includes the name of the user
- **Domain**
  Includes the CA Single Sign-On domain
- **Realm**
  Includes the CA Single Sign-On realm
- **AgentName**
  Includes the Agent name being used
- **TransactionID**
  Includes the transaction ID
- **DomainOID**
  Includes the CA Single Sign-On domain OID
- **IPAddr**
  Includes the client IP address
- **RequestIPAddr**
  Includes the trace file displays the IP of the server where Agent is present
- **IPPort**
  Includes the client IP port

- **CertSerial**
  Includes the certificate serial number
- **SubjectDN**
  Includes the subject DN of the certificate
- **IssuerDN**
  Includes the Issuer DN of the certificate
- **SessionSpec**
  Includes the CA Single Sign-On session spec
- **SessionID**
  Includes the CA Single Sign-On session ID
- **UserDN**
  Includes the User DN
- **Resource**
  Includes the requested resource
- **Action**
  Includes the requested action
- **RealmOID**
  Includes the realm OID
- **ResponseTime**
  Includes the average response time in milliseconds of the Policy Servers associated with a CA Web Agent or SDK Agent and API application
  **Note:** To output the ResponseTime to a trace log, include the component Agent_Con_Manager along with the data field ResponseTime in the WebAgentTrace.conf file or other file specified in the Policy Server Configuration Object (ACO) and restart the Policy Server. The Agent_Con_Manager component, or Agent API Connection Manager, calculates the ResponseTime each time a response is received from a Policy Server and keeps a running average. To locate the ResponseTime in the trace log, search for [PrintStats].

## Trace Message Data Field Filters

To focus on a specific problem, you can narrow the output of the trace log by specifying a filter based on the value of a data field. For example, if you are having problems with an index.html page, you can filter on resources with an html suffix by specifying Resource:==/html in the trace configuration file. Each filter should be on a separate line in the file.

Filters use the following syntax:

*data_field*:*filter*

The following types of filters are available:

- == (exact match)

- != (does not equal)

The filters use boolean logic as shown in the following examples:

Action:!=get (all actions except get)

Resource:==/html (all resources ending in /html)

## Determine the Content of the Trace Log

The WebAgentTrace.conf file determines the content of the trace log. You can control which components and data items appear in your trace log by modifying the settings of the WebAgentTrace.conf file on your web server. The following factors apply when editing the file:

- Entries are case-sensitive.
  When you specify a component, data field, or filter, the values must match exactly the options in the WebAgentTrace.conf file instructions.
- Uncomment the configuration settings lines.
- If you modify the WebAgentTrace.conf file before installing a new agent over an existing agent, the file is overwritten. Rename or back up the file first. After the installation, you can integrate your changes into the new file.

**Follow these steps:**

1. Open the WebAgentTrace.conf file.

   **Note:** We recommend duplicating the original file and changing the copy. Modifying the copy preserves the default settings.

2. Add components and subcomponents using the following steps:

   a. Find the section that matches your type of agent. For example, if you have an Apache 2.0 Agent that is installed on your server, look for a line resembling the following example:

   ```
   # For Apache 2.0, Apache 2.2, IIS 7.0 and SunOne Web Agents
   ```

   b. Locate the following line in that section:

   ```
   #components:
   ```

   c. Uncomment the line. Then add the component names that you want after the colon. Separate multiple components commas as shown in the following example:

   ```
   components:  AgentFramework, HTTPAgent
   ```

   d. (Optional) Follow the component name with the name of a subcomponent you want. Separate the subcomponent name with a slash as shown in the following example:

   ```
   components:  AgentFramework/Administration
   ```

3. Add data fields and filters using the following steps:

   a. Locate the following line in the appropriate section:

      ```
      #data:
      ```

   b. Uncomment the line. Then add the data fields that you want after the colon. Separate multiple data fields with commas as shown in the following example:

      ```
      data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr
      ```

   c. (Optional) Add filters to your data fields by following the data field with a colon, the Boolean operator and the value you want. The values you specify for the filters must match exactly. The following example shows a filter which logs activities for a specific IP address:

      ```
      data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr:
      ==127.0.0.1
      ```

      **Note:** Each filter must be on a separate line in the file.

4. Save your changes and close the file.

5. Restart the web server to apply your changes.
   The content of the trace log has been determined.

## Limit the Number of Trace Log Files Saved

You can limit the number of trace logs that a CA Single Sign-On agent keeps. For example, if you want to save disk space on the system that stores your agent logs, you can limit the number of trace logs using the following parameter:

TraceFilesToKeep

Specifies the number of CA Single Sign-On agent trace log files that are kept. New trace logs are created in the following situations:

- When the agent starts.
- When the size limit of the trace log (specified by the value of the TraceFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing trace logs which exceed the number that you want to keep. For example, If your system has 500 trace logs stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 trace logs.

Setting the value of this parameter to zero retains all the trace logs.

**Default**: 0

**Follow these steps:**

1. Archive or delete any existing trace logs from your system.
2. Set the value of the TraceAppend parameter to no.
3. Change the value of the TraceFilesToKeep parameter to the number of trace logs that you want to keep.

## Collect Detailed Agent Connection Data with an Agent Connection Manager Trace Log

To collect detailed information about the connections between a CA Single Sign-on WSS Agent and Policy Server, you create a Trace Log file that contains information gathered by the Agent Collection Manager.

**Follow these steps:**

1. Open your Agent Configuration object or local configuration file.
2. Set the value of the TraceFile parameter to yes.

**Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings defined on the Policy Server. For example, when the value of this parameter is set to yes in a LocalConfig.conf file log files are generated even if the value of the AllowLocalConfig parameter in the corresponding Agent Configuration object on the Policy Server is set to no. Additionally, set the related trace logging parameters (that define the file name, size, and so on) in the LocalConfig.conf file to override any Policy Server trace log settings.

3. Specify the full path to the trace log file for your Agent Connection Data in the TraceFileName parameter. This is the file that contains the trace log output.
4. Set the value of the TraceConfigFile parameter to the full path of the following file:

   `agent_home/config/AgentConMgr.conf`

   - **agent_home**
     Indicates the directory where the CA Single Sign-on WSS Agent is installed on your web server.
     **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
     **Default** (Windows 64-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
     **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations operating on 64-bit systems: C:\Program Files (x86)\CA\Web Services Security\webagent\win32
5. Define the format the trace log file for your Agent Connection Data by setting the following parameters:

- **TraceAppend**
  Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
  **Default:** No

- **TraceDelimiter**
  Specifies a custom character that separates the fields in the trace file.
  **Default:** No default
  **Example:** |

- **TraceFileSize**
  Specifies (in megabytes) the maximum size of a trace file. The Web Agent creates a new file when this limit is reached.
  **Default:** 0 (a new log file is not created)
  **Example:** 20 (MB)

- **TraceFormat**
  Specifies how the trace file displays the messages. Choose *one* of the following options:

  - default—uses square brackets [] to enclose the fields.
  - fixed—uses fields with a fixed width.
  - delim—uses a character of your choice to delimit the fields.
  - xml—uses XML-like tags. A DTD or style sheet is *not* provided with the Web Agent.

  **Default:** default (square brackets)

- **LogLocalTime**
  Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
  **Default:** Yes

6. Restart your web server so the new settings take effect.
   Detailed information about the CA Single Sign-on WSS Agent connections is collected.

> **Note**: For this version of CA Single Sign-On the BusyHandleCount and FreeHandleCount attributes are not used.

## Configure XML Message Processing Logging

In addition to Web Agent logging functionality, the CA Single Sign-on WSS Agent provides an additional level of log information relating specifically to its processing of XML messages. CA Single Sign-on WSS Agent logging is implemented using Apache's *log4j* standard (see http://logging.apache.org).

**Note:** CA Single Sign-on WSS Agent logging does not start until an XML message that needs to be processed is received.

By default, CA Single Sign-on WSS Agent logging is enabled and written to the soasm_agent.log file in:

- Windows—*agent_home*\bin\
- UNIX—*agent_home*/bin/

  - **agent_home**
    Indicates the directory where the CA Single Sign-on WSS Agent is installed on your web server.
    **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent
    **Default** (Windows 64-bit CA Single Sign-on WSS Agent installations: C:\Program Files\CA\Web Services Security\webagent\win64
    **Default** (Windows 32-bit CA Single Sign-on WSS Agent installations operating on 64-bit systems: C:\Program Files (x86)\CA\Web Services Security\webagent\win32

You can change logging parameters for your CA Single Sign-on WSS Agent by editing the log.config file, which can be found in:

- Windows—*agent_home*\config\
- UNIX— *agent_home*/config/

### Disable CA Single Sign-on WSS Agent XML Message Processing Logging

To disable CA Single Sign-on WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent config subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

# Web Services Security Agent for Oracle iPlanet Servers

The following sections detail how to install and configure a WSS agent on an Oracle iPlanet web server.

## Hardware Requirements for WSS Agents on Oracle iPlanet Servers

Computers hosting WSS agents require the following hardware:

- **Windows requirements**
  Agents on Windows operating environments require the following hardware:

- CPU: x86 or x64
- Memory: 2-GB system RAM.
- Available disk space:
  - 2-GB free disk space in the installation location.
  - .5-GB free disk space in the temporary location.

- **UNIX requirements**
  Agents on UNIX operating environments require the following hardware:
  - CPU:
    - Solaris operating environment: SPARC
    - Red Hat operating environment: x86 or x64
  - Memory: 2-GB system RAM.
  - Available disk space:
    - 2-GB free disk space in the installation location.
    - .5-GB free disk space in /temp.

    Daily operation of the agent requires 10 MB of free disk space in /tmp. The agent creates files and named pipes under /tmp. The path to which these files and pipes are created cannot be changed.

# Policy Server Requirements for WSS Agents on Oracle iPlanet Servers

Verify the following criteria:

- Your Policy Server is installed and configured.
- Your Policy server can communicate with the computer where you plan to install the agent.

To install and configure a WSS agent, a Policy Server requires at least the following items:

- A CA Single Sign-On administrator that has the right to register trusted hosts.
  A trusted host is a client computer where one or more Agents are installed and registered with the Policy Server. The administrator must have permissions to register trusted hosts with the Policy Server. Registering a trusted host creates a unique trusted host name object on the Policy Server.

- An Agent identity
  An Agent identity establishes a mapping between the Policy Server and the name or IP address of the web server instance hosting an Agent. You define an Agent identity from the Agents object in the Administrative UI. You assign it a name and specify the Agent type as a Web Agent.

- A Host Configuration Object (HCO)
  The host configuration object on the Policy Server defines the communication between the agent and the Policy Server that occurs after an initial connection. The Initial connections use the parameters in the SmHost.conf file.

- Agent Configuration Object (ACO)
  This object includes the parameters that define the agent configuration. All agents require at least one of the following configuration parameters that are defined in the ACO:

- **AgentName**
  Defines the identity of the web agent. This identity links the name and the IP address or FQDN of each web server instance hosting an Agent.
  The value of the DefaultAgentName is used instead of the AgentName parameter if any of the following events occur:

  - The AgentName parameter is disabled.
  - The value of AgentName parameter is empty.
  - The values of the AgentName parameter do *not* match any existing agent object.

  Note: This parameter can have more than one value. Use the multivalue option when setting this parameter in an Agent Configuration Object. For local configuration files, add each value to a separate line in the file.
  **Default:** No default
  **Value**: Multiple values are allowed, but each AgentName parameter has a 4,000 character limit. Create additional AgentName parameters as needed by adding a character to the parameter name. For example, AgentName, AgentName1, AgentName2.
  **Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.
  **Examples:**

  - myagent1,192.168.0.0 (IPV4)
  - myagent2, 2001:DB8::/32 (IPV6)
  - myagent,www.example.com (http://www.example.com)
  - (multiple AgentName parameters): AgentName1, AgentName2, AgentName3. The value of each AgentName*number* parameter is limited to 4,000 characters.

- **DefaultAgentName**
  Defines a name that the agent uses to process requests. The value for DefaultAgentName is used for requests on an IP address or interface when no agent name value exists in the AgentName parameter.

If you are using virtual servers, you can set up your environment quickly by using a DefaultAgentName. Using DefaultAgentName means that you do not need to define a separate agent for each virtual server.

**Important!** If you do not specify a value for the DefaultAgentName parameter, then the value of the AgentName parameter requires every agent identity in its list. Otherwise, the Policy Server cannot tie policies to the agent.

**Default:** No default.
**Value**: Multiple values are allowed.
**Limits:** Must contain 7-bit ASCII characters in the range of 32-127, and include one or more printable characters. Cannot contain the ampersand (&) and asterisk (*) characters. The value is not case-sensitive. For example, the names MyAgent and myagent are treated the same.

# WSS Agent Installation Prerequisites for Oracle iPlanet Web Server

Before installing the WSS agent on a system, complete the following tasks for the system operating environment.

- Windows 64-bit Systems Require a C++ Redistributable Package (see page )
- UNIX Remote Terminal Installations Require the DISPLAY Variable be Set (see page 739)
- Required Solaris Patches (see page 740)
- AIX System Runtime Environment Version (see page 740)
- Required Linux Packages (see page 740)
- Required Linux Libraries (see page 740)

## Windows 64-bit Systems Require a C++ Redistributable Package

On an Oracle iPlanet server running a Windows 64-bit platform, download and install the Microsoft Visual C++ 2005 Redistributable Package (x64). Go to the Microsoft web site (http://www.microsoft.com/), to download the Microsoft Visual C++ 2005 Redistributable Package (x64).

## UNIX Remote Terminal Installations Require the DISPLAY Variable be Set

If you are installing the Agent on a UNIX system from a remote terminal, such as a Telnet or Exceed terminal, set the DISPLAY variable for the local system. For example, if your machine is 111.11.1.12, set the variable as follows:

DISPLAY=111.11.1.12:0.0

export DISPLAY

**Note:** You can also install the agent using the console mode installation, which does not require the X window display mode.

## Required Solaris Patches

Before installing an agent on a Solaris system, install the following patches:

- **Solaris 9**
  Requires patch 111711-16.

- **Solaris 10**
  Requires patch 119963-08.

You can verify installed patch versions by logging in as the root user and executing the following command:

```
showrev -p | grep patch_id
```

To locate Solaris patches, go to the Oracle Solution Center.

## AIX System Runtime Environment Version

To run a rearchitected (framework) agent for Oracle iPlanet on an AIX system, your C/C++ runtime environment must be version 8.0.0.0.

## Required Linux Packages

The following software packages are required for Web Agents running on 64-bit Linux systems

- Binutils 2.17

- GCC 4.1.0

## Required Linux Libraries

CA Single Sign-On requires certain Linux libraries for components that operate on Linux. We recommend using YUM to install the required libraries as YUM resolves the dependencies of packages and their versions.

The following list describes the commands to install the required libraries on the host system:

**Red Hat 5.x**

```
yum install -y compat-gcc-34-c++
yum install -y libidn.i686
yum install -y libstdc++.i686
yum install -y ncurses-libs.i686
```

**Red Hat 6.x**

```
yum install -y libstdc++.i686
yum install -y libidn.i686
yum install -y libXext.i686
yum install -y ncurses-libs.i686
yum install -y libXrender.i686
yum install -y libXtst.i686
```

**Additional Packages for Red Hat 6.x 64-bit**

```
yum install -y libXau.i686
yum install -y libXext.i686
yum install -y libxcb.i686
yum install -y compat-libstdc++-33.i686
yum install -y compat-db42.i686
yum install -y compat-db.i686
yum install -y compat-db43.i686
yum install -y libXi.i686
yum install -y libX11.i686
yum install -y libXtst.i686
yum install -y libXrender.i686
yum install -y libXft.i686
yum install -y libXt.i686
yum install -y libXp.i686
yum install -y libstdc++.i686
yum install -y libICE.i686
yum install -y compat-libtermcap.i686
yum install -y libidn.i686
yum install -y libSM.i686
yum install -y libuuid.i686
```

If the correct library is unavailable, CA Single Sign-On displays the following error:

```
java.lang.UnsatisfiedLinkError
```

# Install and Configure WSS Agents for Oracle iPlanet Servers on Windows

The information in this topic explains how to install and configure the WSS Agent on an Oracle iPlanet Server.



## Tasks to Complete Before Installing the WSS Agent

### Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.
2. Double-click System.

3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.
3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Gather Information for the Installation

Gather the following information about your web server before running the installation program for the agent:

- **Installation Directory**
  Specifies the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location.
  **Limit**: The product requires the name "webagent" for the bottom directory in the path

## Gather Information for the Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Install the WSS Agent

Install the WSS Agent using the installation media on the Technical Support site.

**Follow these steps:**

1. Exit all applications that are running.

2. Navigate to the installation material.

3. Double-click ca-sm-wss-*version*-win32.exe.
   *version* specifies the version and, if applicable, the cumulative release number. The base version does not include a cumulative release number in the file name.

   The installation wizard starts.

   **Important!** If Windows User Account Control (UAC) is enabled, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

4. Go through the installation using gathered system and component information. Consider the following points when running the installer:

   - When prompted to select which Web Services Security Agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables on the system. Select a supported 32-bit Java Runtime Environment (see the Platform Support Matrix on the Technical Support site).
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - If the installer detects an existing Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

5. Review the information that is presented on the Pre-Installation Summary page, then click Install.
   If the installation program detects that newer versions of certain system DLLs are installed on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location.

6. On the Web Services Security Configuration screen, select an option to configure the agent now or later then click Next.
   If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

7. Click Done.

8. Do one of the following:

   - If you selected the option to configure WSS Agents now, the installation program prepares the Web Services Security Configuration Wizard and begins the trusted host registration and configuration process. Use the information that you gathered earlier to complete the wizard.
   - If you did not select the option to configure WSS Agents now, or if you are required to reboot the system after installation, run the configuration wizard manually later.

After installation, you can review the installation log file in *wss_home*\install_config_info. The file name is: CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log

*wss_home* specifies the path to where Web Services Security is installed. The default location is C:\Program Files\CA\Web Services Security

*install-date-and-time* specifies the date and time that the WSS Agent was installed.

The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Configure the WSS Agent

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

**Follow these steps:**

1. Open the following directory on your web server:

   *WSS_Home*\install_config_info

   - ***WSS_Home***
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, right-click ca-pep-config.exe, and then select Run as Administrator:

   - For a console-based configuration, enter the following command from a Command Prompt window with Administrator privileges open to *WSS_Home*\install_config_info:

     ```
     ca-pep-config.exe -i console
     ```

     **Important!** If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

3. Use the information you gathered earlier to complete the wizard.
   The agent runtime instance is created for your web servers.

## Apply WSS Agent Changes to Oracle iPlanet obj.conf File (SunOne 6.1 Servers Only)

The Agent Configuration Wizard modifies the default obj.conf, and mime.types files that the Oracle iPlanet web server uses.

If you are using version 6.1 of a SunOne web server, and you plan to use the Oracle iPlanet Administration console, apply the changes to these files *before* using the console. If you do not apply the changes before using the console, the configuration changes can be corrupted. If you lose your configuration, run the configuration program again.

**Follow these steps:**

1. Log in to the Oracle iPlanet Administration Server console.
2. From the Servers tab, select the web server with the agent installed and click Manage.
3. In the right corner of the dialog, click Apply.
   A warning message about loading the modified configuration files appears.
4. Click Load Configuration Files.
5. Exit the console.
6. Restart the web server.
7. Optimize the Agent for Oracle iPlanet by tuning the shared memory segments.
   The changes are applied.

**Note:** In addition to changes made by the WSS Agent configuration wizard, if the Agent is configured to support an advanced authentication scheme, it adds settings to the Oracle iPlanet obj.conf file. You must remove these obsolete settings by editing the obj.conf file.

## Configure Non-Default Server Instances, Virtual Servers, or Reverse Proxies for Oracle iPlanet Web Servers

The WSS Agent configuration wizard only configures the default instance of your Oracle iPlanet web server. To configure a different instance of the web server, edit the **obj.conf** file that is associated with that server instance. Examples of server instances that need configuration include:

- Servers installed in a nondefault directory
- Servers that you want to configure as a reverse proxy server. We recommend configuring the reverse proxy using your Oracle iPlanet interface *before* editing the obj.conf file.
  The Agent configuration wizard *only* modifies the *default* obj.conf file on the Oracle iPlanet web server. To protect other instances or reverse proxy deployments, copy the settings from the default obj.conf file to any respective *instance_name*-obj.conf files. For example, your web server created an obj.conf file when you installed it, but later you added a server instance named my_server.example.com. To protect resources on my_server.example.com, copy the settings the wizard added from the obj.conf file to the my_server.example.com-obj.conf file.
- Virtual servers on the same computer

**Note**: SunOne/Sun Java 7.0 web servers do *not* require these manual configuration steps.

**Follow these steps to edit the obj.conf file:**

1. Locate the directory of the server instance you want to configure.

2. Open the obj.conf file with a text editor.

3. Locate the following line:

   ```
   <Object name="default">
   ```

4. Insert a new line below the previous one, and then add the following text:

   ```
   AuthTrans fn="SiteMinderAgent"
   ```

5. Locate the following line:

   ```
   AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
   ```

6. Insert a new line below the previous one, and then add the following text:

   ```
   NameTrans fn="pfx2dir" from="/siteminderagent/pwcgi" dir="agent_home/pw" name="
   cgi"
   NameTrans fn="pfx2dir" from="/siteminderagent/pw" dir="agent_home/pw"
   NameTrans fn="pfx2dir" from="/siteminderagent/jpw" dir="agent_home/jpw"
   NameTrans fn="pfx2dir" from="/siteminderagent/redirectjsp" dir="agent_home
   /affwebservices/redirectjsp"
   NameTrans fn="pfx2dir" from="/siteminderagent/certoptional" dir="agent_home
   /samples"
   NameTrans fn="pfx2dir" from="/siteminderagent" dir="agent_home/samples"
   NameTrans fn="pfx2dir" from="/siteminderagent/pwservlet" dir=agent_home/jpw"
   ```

   *agent_home* indicates the directory where the WSS Agent is installed on your web server. Default installation locations:

   **Windows 32-bit:** C:\Program Files\CA\Web Services Security\webagent
   **Windows 64-bit:** C:\Program Files\CA\Web Services Security\webagent\win64
   **Windows 32-bit installations operating on 64-bit systems:** C:\Program Files(x86)\CA\Web Services Security\webagent\win32

7. Locate the following line:

   ```
   NameTrans fn="ntrans-j2ee" name="j2ee"
   ```

8. Insert a new line below the previous one, and then add the following text:

   ```
   PathCheck fn="SmRequireAuth"
   ```

9. Remove the following line:

   ```
   NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Program Files/Sun/WebServer7.0
   /lib/icons" name="es-internal"
   ```

10. Locate the following line:

    ```
    ObjectType fn="force-type" type="text/plain"
    ```

11. Insert a new line below the previous one, and then add the following text:

    ```
    Service method="(GET|POST)" fn="SmAdvancedAuth"
    ```

12. Locate the following line:

    ```
    Error fn="error-j2ee
    ```

13. Insert a new line above the previous one, and then add the following text:

```
Error fn="SmSoapFault" code="500" reason="SmSoapFault"
```

14. Save the obj.conf file.

15. Open the magnus.conf file with a text editor.

16. Add the following line:

```
Init fn="load-modules" shlib="agent_home/bin/SunOneWebAgent.dll" funcs="
SmInitAgent,SmInitChild,SiteMinderAgent,SmRequireAuth,SmAdvancedAuth,SmSoapFault
```

17. Save the magnus.conf file.

The Oracle iPlanet web server is manually configured.

## (Optional) Run the Unattended Installation and Configuration for Additional WSS Agents

After you install a component the first time, you can install the component on other systems using an *unattended installation*. An unattended installation lets you complete the installation without user intervention while the installation executes. This method saves time if you have a large Web Services Security environment that uses many agents with identical settings. For example, if the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers, create your own script to run an unattended installation to install more agents.

After your initial installation, a properties file is also installed. Each component is associated with its own properties file or files. The following guidelines apply to all properties files. Review them before starting an unattended installation:

- Back up the default properties file before modifying it.
- Do not add extra spaces between a parameter name, the equal sign (=), and the parameter value.
- Save the file after you change it.
- Do not manually edit encrypted passwords. These passwords are encrypted for security reasons and cannot be edited in plain text. If you want to add plain text passwords, comment out the encrypted password parameter and uncomment the plain text reference.

**Follow these steps to run an unattended installation:**

1. Run the following wizards on your first web server (in the order shown):

    a. The Web Services Security Installation wizard.

    b. The Web Services Security Configuration wizard.

2. Locate the following file on your first web server:

    *wss_home*/install_config_info/ca-wss-installer.properties

    If the path contains spaces, surround it with quotes.
    *wss_home* specifies the path to where Web Services Security is installed.

3. Copy the properties file to a temporary directory on the new web server.

4. Open the properties file in a text editor and, if necessary, modify the parameters. Save the file. The default parameters in the file reflect the information that was entered during the initial installation.

5. Copy the following files from the first web server to the temporary directory on the new web server:

   a. 
   - The WSS Agent Installation executable file.
   - The ca-pepconfig-installer.properties file.

6. Open a Command Prompt window with root privileges in the temporary directory and run the following command:

   `ca-sm-wss-`*`version`*`-win32.exe -f `*`properties_file`*` -i silent.`

   *version* specifies the version and, if applicable, the cumulative release number. The base version does not include a cumulative release number in the file name.

   **Important!** If Windows User Account Control (UAC) is enabled, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

7. (Optional) Delete the temporary directory from your web server.

8. Repeat Steps 3-7 for each additional web server that you want to add to your environment with the same configuration.

   Note: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

## (Optional) Improve Server Performance with httpd.conf File Changes

You can improve server performance by modifying the default configuration settings in the httpd.conf file; however, these changes are *not* required:

**Follow these steps:**

1. For Oracle iPlanet web servers, assign a higher priority level to your Apache20WebAgent.dll file than any other auth modules or access modules on your web server.

2. For low-traffic websites, define the following directives:

   - Set MaxRequestsPerChild>1000 *or* Set MaxRequestsPerChild=0

- MinSpareServers >5
- MaxSpareServers>10
- StartServers=MinSpareServers>5

3. For high-traffic websites, define the following directives:
   - Set MaxRequestsPerChild>3000 *or* Set MaxRequestsPerChild=0
   - MinSpareServers >10
   - MaxSpareServers>15
   - StartServers=MinSpareServers>10

# Install and Configure WSS Agents for Oracle iPlanet Servers on UNIX Systems

**Contents**



## Complete Tasks Before Installing the Agent

### Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

**To set the JRE in the PATH variable**

1. Open a Command Window.
2. Run the following commands:

```
PATH=$PATH:JRE
export PATH
```

   - *JRE* defines the location of your Java Runtime Environment bin directory.

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider. IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.
3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
```

```
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Gather the Information for the Installation

Before running the agent installation program, determine the location for the installation directory. This directory is the location of the agent binary files on your web server. The *web_agent_home* variable is set to this location. The product requires that the name "webagent" be the final directory in the path.

## Gather Information for the WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:

Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Install the WSS Agent on a UNIX System

You can install the WSS Agent on a UNIX system using a console or a GUI. To install using a console, execute the Option Pack binary with the **-i console** command argument.

### Install a WSS Agent Using a Console

Install the WSS Agent using the Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  `chmod +x ca-sm-wss-`*`version`*`-`*`unix_version`*`.bin`

  - *version*
    Specifies the version and, if applicable, the cumulative release number. The base version does not include a cumulative release number in the file name.

  - *unix_version*
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the Web Services Security installer across different subnets, it can crash. Install Web Services Security components directly on the host system to avoid the problem.

**Follow these steps:**

1. Exit all applications that are running.

2. Open a shell and navigate to where the install program is located.

3. Enter the following command:

   ```
   ./ca-sm-wss-version-unix_version.bin -i console
   ```

   The Web Services Security installer starts.

4. Use gathered system and component information to install the WSS Agent. Consider the following as you make your selections:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - Do not uses spaces in the WSS Agent install path.
   - If the installer detects the presence of an existing Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

5. Review the information presented on the Pre-Installation Summary page, then proceed.

   **Note:** If the installation program detects newer versions of certain system libraries on your system, it asks if you want to overwrite these newer files with older files. If you see this message, select No To All.

   The WSS Agent files are copied to the specified location. Afterward the configuration screen is displayed.

6. Do one of the following:

   - If you selected the option to configure WSS Agents now, the installation program prepares the Web Services Security Configuration Wizard and begins the trusted host registration and configuration process. Use the information that you gathered earlier to complete the wizard.
   - If you did not select the option to configure WSS Agents now, or if you are required to reboot the system after installation, run the configuration wizard manually later.

To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file. The log file is in the directory *wss_home/*install_config_info.

- *wss_home s*pecifies the path to where Web Services Security is installed.

- *install-date-and-time* specifies the date and time that the WSS Agent was installed.

The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Install the WSS Agent Using a GUI

Install the WSS Agent using the Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  `chmod +x ca-sm-wss-`*`version-unix_version`*`.bin`

  - *version*
    Specifies the version and, if applicable, cumulative release number. The base version does not include a cumulative release number in the file name.

  - *unix_version*
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the Web Services Security installer across different subnets, it can crash. Install Web Services Security components directly on the host system to avoid the problem.

**Follow these steps:**

1. Exit all applications that are running.

2. Open a shell and navigate to where the install program is located.

3. Enter the following command:

   `./ca-sm-wss-`*`version-unix_version`*`.bin`

   The Web Services Security installer starts.

4. Use gathered system and component information to install the WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agent for Web Servers**

   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (see the Platform Support Matrix on the Technical Support site).

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

   - If the installer detects the presence of an existing Web Agent, it displays a warning dialog stating that the install will upgrade the Web Agent. Click Continue to upgrade the Web Agent to a WSS Agent. If you proceed, the software upgrade occurs in the installed location of the existing Web Agent.

- Do *not* use spaces in the WSS Agent install path.

5. Review the information presented on the Pre-Installation Summary page, then click Install.
   If the installation program detects that newer versions of certain system libraries on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location. Afterward, the CA Web Services Security Configuration screen is displayed.

6. Select one of the following options:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

7. Click Done.
   If you selected the option to configure WSS Agents now, the installation program prepares the Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.

   If you did not select the option to configure WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file. The log is the directory *wss_homeE/*install_config_info.

- *wss_home* specifies the path to where Web Services Security is installed.
- *install-date-and-time* specifies the date and time that the WSS Agent was installed.

The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Set Environment Variables for a WSS Agent on UNIX

After installing the WSS Agent on UNIX, you must set required environment variables using the ca_wa_env.sh script. Running the script for WSS Agents on most UNIX platforms ensures that the WSS Agent and web server can work together.

The ca_wa_env.sh script sets the following environment variables:

- NETE_WA_ROOT
- PATH
- NETE_WA_PATH
- LD_LIBRARY_PATH
  The WSS Agent requires that LD_LIBRARY_PATH include /usr/lib before any other directory containing older versions of libm.so (http://libm.so).
- SHLIB_PATH

- LIBPATH

To set the WSS Agent environment variables after installation, source the following script after you install and configure the WSS Agent:

1. Open a command window.
2. Navigate to *WSS_Home*/webagent/.
   *WSS_Home* specifies the path to where Web Services Security is installed.
3. Enter the following command:
   . ./ca_wa_env.sh

**Note:** You do not have to run this script for Sun Java System web servers because this file as been added to the start script.

## Run the WSS Agent Configuration Program

You can configure your WSS Agents and register a trusted host immediately after installing the WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a WSS Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.
- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

**To configure Agents and register a trusted host**

1. If necessary, start the Configuration Wizard as follows:
   a. Open a console window.
   b. Navigate to *agent_home*/install_config_info, where *agent_home* is the installed location of the WSS Agent.

c. Enter one of the following commands:
GUI Mode: ./ca-pep-config.bin
Console Mode: ./ca-pep-config.bin -i console

The Configuration Wizard starts.

2. Use gathered system and component information to configure the WSS Agent and register the host.
If you choose to configure multiple Agents, you can set the register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*/config. You can modify this file.

- *agent_home* is the installed location of the WSS Agent

## Apply WSS Agent Changes to Oracle iPlanet Configuration Files (for SunOne 6.1 Servers only)

The Agent Configuration Wizard modifies the default obj.conf, and mime.types files that the Oracle iPlanet web server uses.

If you are using version 6.1 of a SunOne web server, and you plan to use the Oracle iPlanet Administration console, apply the changes to these files *before* using the console. If you do not apply the changes using the console first, the changes that are made for your configuration could be corrupted. If you lose your configuration, run the configuration program again.

**Note:** The agent adds settings to the obj.conf file of the Oracle iPlanet web server when the Agent is configured to support an advanced authentication scheme. CA Single Sign-On does *not* remove these settings later. Edit the obj.conf file manually to remove any obsolete settings.

**Follow these steps:**

1. Log in to the Oracle iPlanet Administration Server console.
2. From the Servers tab, select the web server with the WSS Agent installed and click Manage.
3. In the right corner of the dialog, click Apply.
A warning message about loading the modified configuration files appears.
4. Click Load Configuration Files.
5. Exit the console.
6. Restart the web server.

7. Optimize the Agent for Oracle iPlanet by tuning the shared memory segments.
   The changes are applied.

## Configure Non-Default Server Instances, Virtual Servers, or Reverse Proxies for Oracle iPlanet Web Servers

The WSS Agent configuration wizard only configures the default instance of your Oracle iPlanet web server. To configure a different instance of the web server, edit the **obj.conf** file that is associated with that server instance. Examples of server instances that need configuration include:

- Servers installed in a nondefault directory
- Servers that you want to configure as a reverse proxy server. We recommend configuring the reverse proxy using your Oracle iPlanet interface *before* editing the obj.conf file.
  The Agent configuration wizard *only* modifies the *default* obj.conf file on the Oracle iPlanet web server. To protect other instances or reverse proxy deployments, copy the settings from the default obj.conf file to any respective *instance_name*-obj.conf files. For example, your web server created an obj.conf file when you installed it, but later you added a server instance named my_server.example.com. To protect resources on my_server.example.com, copy the settings the wizard added from the obj.conf file to the my_server.example.com-obj.conf file.
- Virtual servers on the same computer

**Note**: SunOne/Sun Java 7.0 web servers do *not* require these manual configuration steps.

**Follow these steps to edit the obj.conf file:**

1. Locate the directory of the server instance you want to configure.
2. Open the obj.conf file with a text editor.
3. Locate the following line:

   ```
   <Object name="default">
   ```

4. Insert a new line below the previous one, and then add the following text:

   ```
   AuthTrans fn="SiteMinderAgent"
   ```

5. Locate the following line:

   ```
   AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
   ```

6. Insert a new line below the previous one, and then add the following text:

   ```
   NameTrans fn="pfx2dir" from="/siteminderagent/pwcgi" dir="agent_home/pw" name="
   cgi"
   NameTrans fn="pfx2dir" from="/siteminderagent/pw" dir="agent_home/pw"
   NameTrans fn="pfx2dir" from="/siteminderagent/jpw" dir="agent_home/jpw"
   NameTrans fn="pfx2dir" from="/siteminderagent/redirectjsp" dir="agent_home
   /affwebservices/redirectjsp"
   NameTrans fn="pfx2dir" from="/siteminderagent/certoptional" dir="agent_home
   ```

```
/samples"
NameTrans fn="pfx2dir" from="/siteminderagent" dir="agent_home/samples"
NameTrans fn="pfx2dir" from="/siteminderagent/pwservlet" dir=agent_home/jpw"
```

*agent_home* indicates the directory where the WSS Agent is installed on your web server. Default installation locations:

**Windows 32-bit:** C:\Program Files\CA\Web Services Security\webagent
**Windows 64-bit:** C:\Program Files\CA\Web Services Security\webagent\win64
**Windows 32-bit installations operating on 64-bit systems:** C:\Program Files(x86)\CA\Web Services Security\webagent\win32

7. Locate the following line:

   ```
   NameTrans fn="ntrans-j2ee" name="j2ee"
   ```

8. Insert a new line below the previous one, and then add the following text:

   ```
   PathCheck fn="SmRequireAuth"
   ```

9. Remove the following line:

   ```
   NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Program Files/Sun/WebServer7.0
   /lib/icons" name="es-internal"
   ```

10. Locate the following line:

    ```
    ObjectType fn="force-type" type="text/plain"
    ```

11. Insert a new line below the previous one, and then add the following text:

    ```
    Service method="(GET|POST)" fn="SmAdvancedAuth"
    ```

12. Locate the following line:

    ```
    Error fn="error-j2ee
    ```

13. Insert a new line above the previous one, and then add the following text:

    ```
    Error fn="SmSoapFault" code="500" reason="SmSoapFault"
    ```

14. Save the obj.conf file.

15. Open the magnus.conf file with a text editor.

16. Add the following line:

    ```
    Init fn="load-modules" shlib="agent_home/bin/SunOneWebAgent.dll" funcs="
    SmInitAgent,SmInitChild,SiteMinderAgent,SmRequireAuth,SmAdvancedAuth,SmSoapFault
    ```

17. Save the magnus.conf file.

The Oracle iPlanet web server is manually configured.

## Modify the Oracle iPlanet Startup Script to Prevent Crashes when the Server Stops

The Oracle iPlanet server can sometimes crash when shutting down in the following operating environments:

- Solaris 9 SP3
- Solaris 10

Modify the startserv script to prevent the Oracle iPlanet web server from crashing when shuttng down.

**Follow these steps:**

1. Open the following file with a text editor:

   ```
   sunone_instance_directory/bin/startserv
   ```

   - *sunone_instance_directory* indicates the directory of the SunOne web server instance.

2. Locate the following line:

   ```
   LIBUMEM_32=/usr/lib/libumem.so
   ```

3. Add a comment character in the beginning of the previous line. See the following example:

   ```
   #LIBUMEM_32=/usr/lib/libumem.so
   ```

4. Locate the following line:

   ```
   LIBUMEM_64=/usr/lib/64/libumem.so
   ```

5. Add a comment character in the beginning of the previous line. See the following example:

   ```
   #LIBUMEM_64=/usr/lib/64/libumem.so
   ```

6. Save the file and close the text editor.
   The Oracle iPlanet startup script is modified.

## (Optional) Run an Unattended Installation and Configuration Programs for your WSS Agent

After you install a component the first time, you can install the component on other systems using an *unattended installation*. An unattended installation lets you complete the installation without user intervention while the installation executes. This method saves time if you have a large Web Services Security environment that uses many agents with identical settings. For example, if the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers, create your own script to run an unattended installation to install more agents.

After your initial installation, a properties file is also installed. Each component is associated with its own properties file or files. The following guidelines apply to all properties files. Review them before starting an unattended installation:

- Back up the default properties file before modifying it.
- Do not add extra spaces between a parameter name, the equal sign (=), and the parameter value.
- Save the file after you change it.

- Do not manually edit encrypted passwords. These passwords are encrypted for security reasons and cannot be edited in plain text. If you want to add plain text passwords, comment out the encrypted password parameter and uncomment the plain text reference.

**Follow these steps to run an unattended installation:**

1. Run the following wizards on your first web server (in the order shown):
   a. The Web Services Security Installation wizard.
   b. The Web Services Security Configuration wizard.
2. Locate the following file on your first web server:

   `wss_home/install_config_info/ca-wss-installer.properties`

   If the path contains spaces, surround it with quotes.
   *wss_home* specifies the path to where Web Services Security is installed.
3. Copy the properties file to a temporary directory on the new web server.
4. Open the properties file in a text editor and, if necessary, modify the parameters. Save the file. The default parameters in the file reflect the information that was entered during the initial installation.
5. Copy the following files from the first web server to the temporary directory on the new web server:
   a.
   - The WSS Agent Installation executable file.
   - The ca-pepconfig-installer.properties file.
6. Open a Command Prompt window with root privileges in the temporary directory and run the following command

   `ca-sm-wss-version-unix_version.bin -f ca-wss-insaller.properties -i silent`
7. (Optional) Delete the temporary directory from your web server.
8. Repeat Steps 3-7 for each additional web server that you want to add to your environment with the same configuration.

Note: To automate this process, create your own customized script to execute these files on your systems. Use any scripting language that you want.

# Uninstall a WSS Agent from Oracle iPlanet Servers

Uninstall a WSS Agent to remove the agent from your system.

## Set the JRE in the PATH Variable Before Uninstalling

On Windows and UNIX systems, *before* you uninstall the WSS Agent, set the JRE is in the PATH variable. Complete this step or the uninstallation program stops and issues one of the following error messages:

- Could not find a valid Java virtual machine to load. You need to reinstall a supported Java virtual machine.
- No Java virtual machine could be found from your PATH environment variable. You must install a VM prior to running this program.

**Follow these steps:**

**Windows**

1. Go to the Control Panel.
2. Double-click System.
3. In the Environment Variables dialog, add the location of the JRE to the PATH system variable. For example, C:\j2sdk*version_number*\jre\bin

**UNIX**
Run the following commands:

1. PATH=$PATH:*JRE*/bin
   *JRE* specifies the location of your JRE. For example, /usr/bin/j2sdk*version_number*/jre
2. export PATH

## Uninstall a WSS Agent

To uninstall a WSS Agent, run the Web Services Security uninstall wizard.

**Follow these steps:**

1. Navigate to the following directory for your system:
   Windows: *WSS_HOME*\install_config_info
   UNIX: *WSS_HOME*/install_config_info

   The uninstall wizard starts.
2. Run the Web Services Security uninstall wizard to remove core WSS components:
   Windows: wss-uninstall.cmd
   UNIX: wss-uninstall.sh
   *WSS_HOME* specifies the Web Services Security installation location.

**Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator.

3. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed. If you chose to uninstall only specific features, select the installed components then proceed.
   The uninstall wizard removes all selected Web Services Security components.

4. Restart the server.

The removal of the WSS Agent is complete.

# SiteMinder WSS Agent Logging for Oracle iPlanet Servers

**Contents**



## Logs of Start-up Events

To assist in debugging, startup events are recorded in a log. Each message may provide clues about the problem. These logs are stored in the following locations:

- On Windows systems, these events are recorded in the Windows Application Event log.
- On UNIX systems, these events are sent to STDERR. Apache servers map STDERR to the Apache error_log file, so these events are also recorded in that log.

## Error Logs and Trace Logs

You can use the WSS agent logging function to monitor the performance of the WSS agent and its communication with the Policy Server. The logging feature provides accurate and comprehensive information about the operation of CA Single Sign-On processes to analyze performance and troubleshoot issues.

A log is a record of events that occur during program execution. A log consists of a series of log messages, each one describing some event that occurred during program execution. Log messages are written to log files.

**Note**: IIS Agents create log files only after the first user request is submitted. Apache 2.0 WSS agents create log files when the Apache server starts.

The WSS agent uses the following log files:

- **Error log**
  Contains program and operational-level errors. One example is when the WSS agent cannot communicate with Policy Server. The level of detail output in this log cannot be customized. Error logs contain the following types of messages:

  - **Error messages**
    Contain program-level errors, which indicate incorrect or abnormal program behavior, or an inability to function as expected due to some external problem, such as a network failure. There are also operational-level errors. This type of error is a failure that prevents the operation from succeeding, such as opening a file or authenticating a user.

  - **Informational messages**
    Contain messages for the user or administrator that some event has occurred; that is, that a server has started or stopped, or that some action has been taken.

  - **Warning messages**
    Contain warnings for the user or administrator of some condition or event that is unusual or indicative of a potential problem. This does not necessarily mean there is anything wrong.

- **Trace log**
  Contains detailed warning and informational messages, which you can configure. Examples include trace messages and flow state messages. This file also includes data such as header details and cookie variables. Trace logs contain the following messages:

  - **Trace messages**
    Provide detailed information about program operation for tracing and/or debugging purposes. Trace messages are ordinarily turned off during normal operation. In contrast to informational, warning, and error messages, trace messages are embedded in the source code and can not easily be localized. Moreover, trace messages may include significant data in addition to the message itself; for example, the name of the current user or realm.

You specify the location of both the error and trace log files when you configure the WSS agent. Use the error and trace logs to help solve any issues that may prevent the WSS agent from operating properly.

**Note:** For Agents on Windows platforms, set the EnableWebAgent parameter to yes to ensure that the WSS agent log gets created. If you leave EnableWebAgent set to no (the default) and set the logging parameters, the Agent log gets created only for Agents on UNIX platforms.

## Parameter Values Shown in Log Files

WSS agents list configuration parameters and their values in the WSS agent error log file, but there are differences between the ways that Traditional and Framework agents do this.

Framework agents record the configuration parameters and their values in the log file exactly as you entered them in the Agent Configuration Object or the local configuration file. All of the parameters, including those which may contain an incorrect value, are recorded in the log file.

Traditional agents process the parameter values before recording them. If the parameter has a proper value, the parameter and its value are recorded in the log file. Parameters with incorrect values are *not* recorded in the log file.

## Set Up and Enable Error Logging

Error logs require the following settings:

- Logging is enabled.
- A location for the log file is specified.

The parameters that enable error logging and determine options such as appending log data are defined in a local configuration file or an Agent Configuration Object at the Policy Server.

Agents that are installed on an IIS or Apache web servers do not support dynamic configuration of log parameters that are set locally in a local configuration file. The changes take effect when the Agent is restarts. However, these log settings can be stored and updated dynamically in an agent configuration object at the Policy Server.

**Note:** IIS Agents create log files only after the first user request is submitted. Apache 2.0 WSS agents create log files when the Apache server starts.

**Follow these steps:**

1. If you do not have a log file already, create a log file and any related directories.
2. Set the value of the LogFile parameter to yes.

**Note**: Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig. conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.

3. Specify the full path to the error file, including the file name, in any of the following parameters:

- **LogFileName**
  Specifies the full path (including the file name) of the log file.
  **Default:** No
  **Example:** (Windows) *agent_home*\log\WebAgent.log
  **Example:** (UNIX/LInux) /export/iPlanet/servers/https-jsmith/logs/WebAgent.log

- **LogFileName32**
  Specifies the full path of a log file for a CA Single Sign-on WSS Agent for IIS (on 64-bit Windows operating environments protecting 32-bit applications). The 32-bit applications run in Wow64 mode on the 64-bit Windows operating environment. If logging is enabled but this parameter is not set, the CA Single Sign-on WSS Agent for IIS appends _32 to the log file name.
  **Default**: No
  **Limits**: For Windows 64-bit operating environments only. Specify the file name at the end of the path.
  **Example**: (Windows 64-bit operating environments using Wow64 mode) *agent_home* \log\WebAgent32.log.

4. (Optional) Set the following parameters (in the Agent Configuration Object on the Policy Server or in the local configuration file):

   - **LogAppend**
     Adds new log information to the end of an existing log file. When this parameter is set to no, the entire log file is rewritten each time logging is invoked.
     **Default:** No

   - **LogFileSize**
     Specifies the size limit of the log file in megabytes. When the current log file reaches this limit, a new log file is created. The new log file uses one of the following naming conventions:

     - For framework agents, the new log file has a sequence number that is appended to the original name. For example, a log file named myfile.log is renamed to myfile.log.1 when the size limit is reached.

     - For traditional agents, the new log files are named by appending the date and timestamp to the original name. For example, a log file named myfile.log, is renamed to myfile.log.09-18-2003-16-07-07 when the size limit is reached.

     Archive or remove the old files manually.
     **Default:** 0 (no rollover)
     **Example:** 80

   - **LogLocalTime**
     Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
     **Default:** Yes
     If you use a local configuration file, your settings resemble the following example:

```
LogFile="yes"
LogFileName="/export/iPlanet/servers/https-myserver/logs/errors.log"
LogAppend="no"
LogFileSize="80"
LogLocalTime="yes"
```

Error logging is enabled.

## Enable Transport Layer Interface (TLI) Logging

When you want to examine the connections between the agent and the Policy Server, enable transport layer interface logging.

**To enable TLI logging**

1. Add the following environment variable to your web server.

   ```
   SM_TLI_LOG_FILE
   ```

2. Specify a directory and log file name for the value of the variable, as shown in the following example:

   ```
   directory_name/log_file_name.log
   ```

3. Verify that your agent is enabled.

4. Restart your web server.
   TLI logging is enabled.

## Limit the Number of Log Files Saved

You can limit the number of log files that an agent keeps. For example, if you want to save disk space on the system that stores your agent logs, you can limit the number of log files using the following parameter:

LogFilesToKeep

Specifies the number of agent log files that are kept. New log files are created in the following situations:

- When the agent starts.
- When the size limit of the log file (specified by the value of the LogFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing logs files which exceed the number that you want to keep. For example, If your system has 500 log files stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 files.

Setting the value of this parameter to zero retains all the log files.

**Default**: 0

**Follow these steps:**

Archive or delete any existing log files from your system.

Set the value of the LogAppend parameter to no.

Change the value of the LogFilesToKeep parameter to the number of log files that you want to keep.

# How to Set Up Trace Logging

To set up trace logging, use the following process:

1. Set up and Enable Trace logging.
2. Determine what you want to record in the trace log by reviewing the following lists:
   - Trace Log Components and Subcomponents
   - Trace Message Data Fields
   - Data Field Filters
3. Duplicate the default Trace Configuration File.
4. Modify the duplicate file to include the items you want to record.
5. Restart the agent.

## Configure Trace Logging

Before you can use trace logging, you must configure it by specifying a name, location, and parameters for the trace log file. These settings control the size and format of the file itself. After trace logging is configured, you determine the content of the trace log file separately. This lets you change the types of information contained in your trace log at any time, without changing the parameters of the trace log file itself.

**Follow these steps:**

1. Locate the WebAgentTrace.conf file on your web server. Duplicate the file.
2. Open your Agent Configuration Object or local configuration file.
3. Set the TraceFile parameter to yes.
   Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings that are defined on the Policy Server. For example, suppose that the value of this parameter is set to yes in a LocalConfig.conf file. The agent creates log files even though the value of the AllowLocalConfig parameter in the corresponding agent configuration object is set to no. You can also set the related logging parameters in the LocalConfig.conf file also to override any other settings in the agent configuration object.
4. Specify the full path to the trace log files in following parameters:
   - **TraceFileName**
     Specifies the full path to the trace log file.
     **Default:** No default
     Value: Specify the file name in this parameter. Example: *agent_home*\log\trace.log
5. Specify the full path to the duplicate copies of WebAgentTrace.conf file (you created in Step 1) in the following parameters:

- **TraceConfigFile**
  Specifies the location of the WebAgentTrace.conf configuration file that determines which components and events to monitor.
  **Default**: No default
  **Example**: *agent_home*\config\WebAgentTrace.conf
  This file is not used until the web server is restarted.

6. Define the format of the information in your trace log file by setting the following parameters in your Agent Configuration Object or local configuration file:

   - **TraceAppend**
     Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
     **Default**: No

   - **TraceFormat**
     Specifies how the trace file displays the messages. Choose one of the following options:

     - default—uses square brackets [] to enclose the fields.
     - fixed—uses fields with a fixed width.
     - delim—uses a character of your choice to delimit the fields.
     - xml—uses XML-like tags. A DTD or style sheet is not provided with the WSS agent.

     **Default**: default (square brackets)

   - **TraceDelimiter**
     Specifies a custom character that separates the fields in the trace file.
     **Default**: No default
     **Example**: |

   - **TraceFileSize**
     Specifies (in megabytes) the maximum size of a trace file. The WSS agent creates a new file when this limit is reached.
     **Default:** 0 (a new log file is not created)
     **Example**: 20 (MB)

   - **LogLocalTime**
     Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
     **Default**: Yes

7. Edit the WebAgentTrace.conf file to have agent monitor the activities you want.
   Framework Agents do not support dynamic configuration of log parameters set locally in the Agent configuration file. Consequently, when you modify a parameter, the change does not take effect until you restart the web server. However, these log settings can be stored and updated dynamically if you configure them in an Agent configuration object on the Policy Server.

8. Restart the web server so the agent uses the new trace configuration file.

## Trace Log Components and Subcomponents

The WSS Agent can monitor specific components. When you monitor a component, all of the events for that component are recorded in the trace log. Each component has one or more subcomponents that the agent can also monitor. If you do not want the agent to record all of the events for a component, you can specify only those subcomponents you want to monitor instead.

For example, if you want to record only the single sign-on messages for an agent on a web server, you would specify the Agent component and the SSO subcomponent.

The following components and subcomponents are available:

- **AgentFramework**
  Records all Agent framework messages. (Applies only to framework agents.) The following subcomponents are available:
  - Administration
  - Filter
  - HighLevelAgent
  - LowLevelAgent
  - LowLevelAgentWP
- **AffiliateAgent**
  Records WSS agent messages related to the 4.x Affiliate Agent, which is part of Federation Security Services, a separately-purchased product. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **SAMLAgent**
  WSS agent messages related to the SAML Affiliate Agent. (Applies only to framework agents.) The following subcomponent is available:
  - RequestProcessing
- **WebAgent**
  Records all WSS agent log messages. Applies to all Agents *except* IIS 6.0 or Apache 2.0 Agents. The following subcomponents are available:
  - AgentCore
  - Cache
  - authentication
  - Responses
  - Management

- SSO
- Filter

- **Agent_Functions**
  Records all Agent API messages. The following subcomponents are available:
  - Init
  - UnInit
  - IsProtected
  - Login
  - ChangePassword
  - Validate
  - Logout
  - Authorize
  - Audit
  - FreeAttributes
  - UpdateAttributes
  - GetSessionVariables
  - SetSessionVariables
  - DeleteSessionVariables
  - Tunnel
  - GetConfig
  - DoManagement
- **Agent_Con_Manager**
  Records messages related to internal processing of the Agent API. The following subcomponents are available:
  - RequestHandler
  - Cluster
  - Server
  - WaitQueue

- Management
- Statistics

## Trace Message Data Fields

You can define what each trace message for a specific component contains by specifying which data fields to include in the message.

Data fields use the following syntax:

```
data:data_field1,data_field2,data_field3
```

Some data fields are shown in the following example:

```
data:message,date,time,user,agentname,IPAddr
```

There may not be data for fields in each message, so blank fields my occur. For example, if you select RealmOID as a data field, some trace messages will display the realm's OID while others will not.

The following data fields are available:

- **Message**
  Includes the actual trace message
- **SrcFile**
  Includes the source file and line number of the trace message
- **Pid**
  Includes the process ID
- **Tid**
  Includes the thread ID
- **Date**
  Includes the date
- **Time**
  Includes the time
- **PreciseTime**
  Includes the time, including milliseconds
- **Function**
  Includes the function in the code containing the trace message
- **User**
  Includes the name of the user
- **Domain**
  Includes the CA Single Sign-On domain
- **Realm**
  Includes the CA Single Sign-On realm

- **AgentName**
  Includes the Agent name being used
- **TransactionID**
  Includes the transaction ID
- **DomainOID**
  Includes the CA Single Sign-On domain OID
- **IPAddr**
  Includes the client IP address
- **RequestIPAddr**
  Includes the trace file displays the IP of the server where Agent is present
- **IPPort**
  Includes the client IP port
- **CertSerial**
  Includes the certificate serial number
- **SubjectDN**
  Includes the subject DN of the certificate
- **IssuerDN**
  Includes the Issuer DN of the certificate
- **SessionSpec**
  Includes the CA Single Sign-On session spec
- **SessionID**
  Includes the CA Single Sign-On session ID
- **UserDN**
  Includes the User DN
- **Resource**
  Includes the requested resource
- **Action**
  Includes the requested action
- **RealmOID**
  Includes the realm OID
- **ResponseTime**
  Includes the average response time in milliseconds of the Policy Servers associated with a CA WSS agent or SDK Agent and API application
  **Note:** To output the ResponseTime to a trace log, include the component Agent_Con_Manager along with the data field ResponseTime in the WebAgentTrace.conf file or other file specified in

the Policy Server Configuration Object (ACO) and restart the Policy Server. The Agent_Con_Manager component, or Agent API Connection Manager, calculates the ResponseTime each time a response is received from a Policy Server and keeps a running average. To locate the ResponseTime in the trace log, search for [PrintStats].

## Trace Message Data Field Filters

To focus on a specific problem, you can narrow the output of the trace log by specifying a filter based on the value of a data field. For example, if you are having problems with an index.html page, you can filter on resources with an html suffix by specifying Resource:==/html in the trace configuration file. Each filter should be on a separate line in the file.

Filters use the following syntax:

*data_field*:*filter*

The following types of filters are available:

- == (exact match)
- != (does not equal)

The filters use boolean logic as shown in the following examples:

Action:!=get (all actions except get)

Resource:==/html (all resources ending in /html)

## Determine the Content of the Trace Log

The WebAgentTrace.conf file determines the content of the trace log. You can control which components and data items appear in your trace log by modifying the settings of the WebAgentTrace.conf file on your web server. The following factors apply when editing the file:

- Entries are case-sensitive.
  When you specify a component, data field, or filter, the values must match exactly the options in the WebAgentTrace.conf file instructions.
- Uncomment the configuration settings lines.
- If you modify the WebAgentTrace.conf file before installing a new agent over an existing agent, the file is overwritten. Rename or back up the file first. After the installation, you can integrate your changes into the new file.

**Follow these steps:**

1. Open the WebAgentTrace.conf file.
2. Duplicate the original file and modify the copy. Modifying the copy preserves the default settings.
3. Add components and subcomponents using the following steps:

a. Find the section that matches your type of agent. For example, if you have an Apache 2.0 Agent that is installed on your server, look for a line resembling the following example:

```
# For Apache 2.0, Apache 2.2, IIS 7.0 and SunOne Web Agents
```

b. Locate the following line in that section:

```
#components:
```

c. Uncomment the line. Then add the component names that you want after the colon. Separate multiple components commas as shown in the following example:

```
components:  AgentFramework, HTTPAgent
```

d. (Optional) Follow the component name with the name of a subcomponent you want. Separate the subcomponent name with a slash as shown in the following example:

```
components:  AgentFramework/Administration
```

4. Add data fields and filters using the following steps:

a. Locate the following line in the appropriate section:

```
#data:
```

b. Uncomment the line. Then add the data fields that you want after the colon. Separate multiple data fields with commas as shown in the following example:

```
data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr
```

c. (Optional) Add filters to your data fields by following the data field with a colon, the Boolean operator and the value you want. The values you specify for the filters must match exactly. The following example shows a filter which logs activities for a specific IP address:

```
data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr:
==127.0.0.1
```

Each filter must be on a separate line in the file.

5. Save your changes and close the file.

6. Restart the web server to apply your changes.
The content of the trace log has been determined.

## Limit the Number of Trace Log Files Saved

You can limit the number of trace logs that an agent keeps. For example, if you want to save disk space on the system that stores your agent logs, limit the number of trace logs using the **TraceFilesToKeep** parameter.

The TraceFilestoKeep parameter specifies the number of agent trace log files that are kept. New trace logs are created in the following situations:

- When the agent starts.

- When the size limit of the trace log (specified by the value of the TraceFileSize parameter) is reached.

Changing the value of this parameter does *not* automatically delete any existing trace logs which exceed the number that you want to keep. For example, If your system has 500 trace logs stored, and you decide to keep only 50 of those files, the agent does *not* delete the other 450 trace logs.

Setting the value of this parameter to zero retains all the trace logs.

**Default**: 0

**Follow these steps:**

1. Archive or delete any existing trace logs from your system.
2. Set the value of the TraceAppend parameter to no.
3. Change the value of the TraceFilesToKeep parameter to the number of trace logs that you want to keep.

## Collect Detailed Agent Connection Data with an Agent Connection Manager Trace Log

To collect detailed information about the connections between a WSS Agent and Policy Server, you create a Trace Log file that contains information gathered by the Agent Collection Manager.

**Follow these steps:**

1. Open your Agent Configuration object or local configuration file.
2. Set the value of the TraceFile parameter to yes.
   Setting the value of this parameter to yes in a local configuration file of a web server overrides any of the logging settings defined on the Policy Server. For example, when the value of this parameter is set to yes in a LocalConfig.conf file log files are generated even if the value of the AllowLocalConfig parameter in the corresponding Agent Configuration object on the Policy Server is set to no. Additionally, set the related trace logging parameters (that define the file name, size, and so on) in the LocalConfig.conf file to override any Policy Server trace log settings.
3. Specify the full path to the trace log file for your Agent Connection Data in the TraceFileName parameter. This is the file that contains the trace log output.
4. Set the value of the TraceConfigFile parameter to the full path of the following file:

   `agent_home/config/AgentConMgr.conf`

   *agent_home* indicates the directory where the WSS Agent is installed on your web server. The default locations are:

   - **Windows 32-bit installations:** C:\Program Files\CA\Web Services Security\webagent
   - **Windows 64-bit installations:** C:\Program Files\CA\Web Services Security\webagent\win64

- **Windows 32-bit installations operating on 64-bit systems:** C:\Program Files (x86)\CA\Web Services Security\webagent\win32\

5. Define the format the trace log file for your Agent Connection Data by setting the following parameters:

1. 
   - **TraceAppend**
     Adds new logging information to the end of an existing log file instead of rewriting the entire file each time logging is invoked.
     **Default:** No

   - **TraceDelimiter**
     Specifies a custom character that separates the fields in the trace file, for example, a pipe character (|).
     **Default:** No default

   - **TraceFileSize**
     Specifies (in megabytes) the maximum size of a trace file. The WSS agent creates a new file when this limit is reached.
     **Default:** 0 (a new log file is not created)
     **Example:** 20 (MB)

   - **TraceFormat**
     Specifies how the trace file displays the messages. Choose *one* of the following options:

     - default—uses square brackets [] to enclose the fields.
     - fixed—uses fields with a fixed width.
     - delim—uses a character of your choice to delimit the fields.
     - xml—uses XML-like tags. A DTD or style sheet is *not* provided with the WSS agent.

     **Default:** default (square brackets)

   - **LogLocalTime**
     Specifies whether the logs use Greenwich Mean Time (GMT) or local time. To use GMT, change this setting to no. If this parameter does not exist, the default setting is used.
     **Default:** Yes

2. Restart your web server so the new settings take effect.
   Detailed information about the WSS Agent connections is collected.

**Note**: For CA Single Sign-On 12.52, the BusyHandleCount and FreeHandleCount attributes are not used.

## Configure XML Message Processing Logging

In addition to WSS agent logging functionality, the WSS Agent provides an additional level of log information relating specifically to its processing of XML messages. WSS Agent logging is implemented using Apache's *log4j* standard (see http://logging.apache.org).

**Note:** WSS Agent logging does not start until an XML message that needs to be processed is received.

By default, WSS Agent logging is enabled and written to the soasm_agent.log file in:

- Windows—*agent_home*\bin\
- UNIX—*agent_home*/bin/
  *agent_home* indicates the directory where the WSS Agent is installed on your web server. The default locations are:
  - Windows 32-bit installations: C:\Program Files\CA\Web Services Security\webagent\
  - Windows 64-bit installations: C:\Program Files\CA\Web Services Security\webagent\win64
  - Windows 32-bit installations operating on 64-bit systems: C:\Program Files (x86)\CA\Web Services Security\webagent\win32

You can change logging parameters for your WSS Agent by editing the log.config file, which can be found in:

- Windows—*agent_home*\config\
- UNIX— *agent_home*/config/

## Disable WSS Agent XML Message Processing Logging

To disable WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent configuration subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

# Web Services Security Agent for Oracle WebLogic

The following sections detail how to install and configure a WSS agent on Oracle WebLogic.

# WSS Agent for Oracle WebLogic Introduction

**Contents**

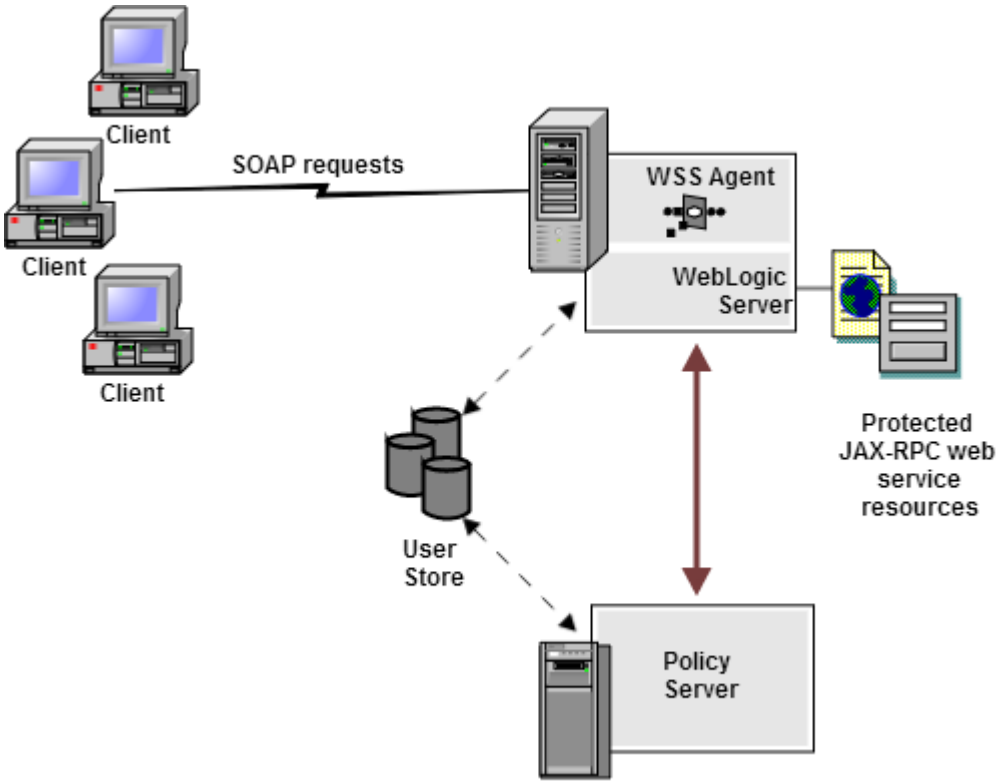


## CA Single Sign-on WSS Agent for Oracle WebLogic Overview

The CA Single Sign-on Web Services Security (WSS) Agent for Oracle WebLogic (formerly SOA Agent) resides in a WebLogic application server, enabling you to protect WebLogic-hosted JAX-RPC web service resources.

The CA Single Sign-on WSS Agent for Oracle WebLogic intercepts all SOAP messages sent over HTTP (S) or JMS transports to JAX-RPC web services deployed on the WebLogic Server. The CA Single Sign-on WSS Agent then communicates with the Policy Server to authenticate and authorize the message sender and, upon successful authentication and authorization, passes the SOAP message on to the addressed web service.

A high-level overview of the CA Single Sign-on WSS Agent for Oracle WebLogic Server architecture is shown in the following figure:



The CA Single Sign-on WSS Agent for Oracle WebLogic provides the following features:

- Fine-grained access control of JAX-RPC web service resources
- Support for bi-directional CA Single Sign-On Web Services Security/CA Single Sign-on and WebLogic single sign-on (SSO)

The CA Single Sign-on WSS Agent additionally supports:

- Multi-byte character user names
- Centralized and dynamic agent configurations
- Caching of resource protection decisions and authentication and authorization decisions
- Logging
- Authorization auditing

## Required Background Information

This section is not intended for users who are new to Java, J2EE standards, or application server technology. It assumes that you have the following technical knowledge:

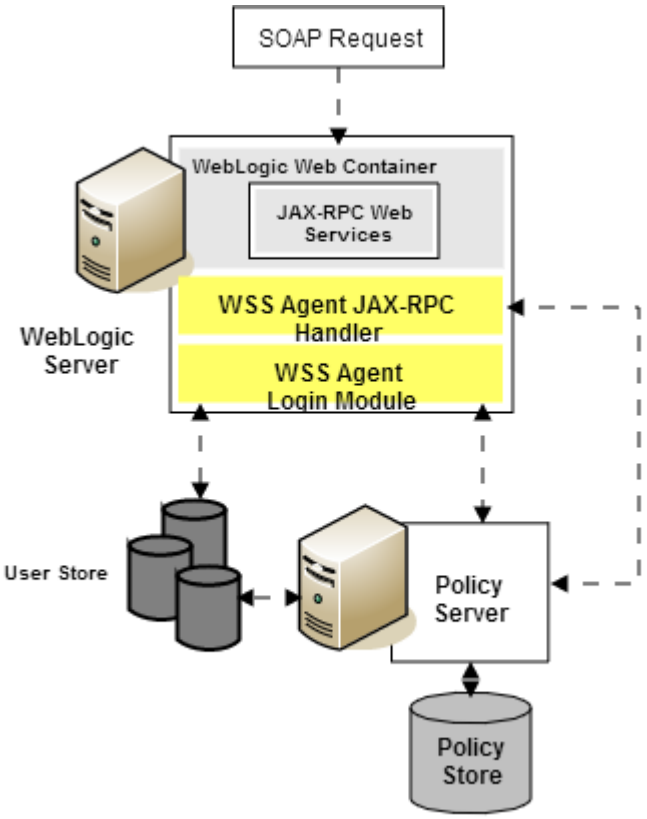
- An understanding of Java, J2EE standards, J2EE application servers, and multi-tier architecture
- An understanding of JAX-RPC web service implementations and JAX-RPC handlers
- Familiarity with the WebLogic Security Framework for WebLogic Server
- Familiarity with Java Authentication and Authorization Server (JAAS) and WebLogic security-related topics
- Experience with managing the WebLogic Server, including tasks such as accessing the administrative console
- Familiarity with CA Single Sign-On Web Services Security concepts, terms, and Policy Server configuration tasks

Additionally, to effectively plan your security infrastructure, you must be familiar with the web services that you plan to protect with CA Single Sign-On Web Services Security.

## CA Single Sign-on WSS Agent for Oracle WebLogic Components

The CA Single Sign-on WSS Agent for Oracle WebLogic consists of two modules that plug into the WebLogic security infrastructure.

- CA Single Sign-on WSS Agent JAX-RPC Handler
- CA Single Sign-on WSS Agent Login Module




## CA Single Sign-on WSS Agent JAX-RPC Handler

The CA Single Sign-on WSS Agent JAX-RPC Handler is a custom JAX-RPC Handler that, when added to the deployment descriptor of a JAX-RPC web service, intercepts SOAP message requests for JAX-RPC web services and diverts them to the CA Single Sign-on WSS Agent Login Module for authentication and authorization decisions.

## CA Single Sign-on WSS Agent Login Module

The CA Single Sign-on WSS Agent Login Module is a JAAS Login Module that performs authentication and authorization for JAX-RPC web services protected by the CA Single Sign-on WSS Agent for Oracle WebLogic.

The CA Single Sign-on WSS Agent Login Module authenticates credentials obtained from the following request types against associated user directories configured in CA Single Sign-On Web Services Security:

- SOAP requests intercepted by the CA Single Sign-on WSS Agent JAX-RPC Handler .
- Requests for web service resources from users with pre-established CA Single Sign-On Web Services Security and CA Single Sign-on sessions (validating the session and obtaining user names from associated CA Single Sign-on session ticket cookies)

If CA Single Sign-On Web Services Security authentication is successful, the CA Single Sign-on WSS Agent Login Module populates a JAAS Subject with a CA Single Sign-On Web Services Security Principal that contains the username and associated CA Single Sign-On Web Services Security session

data. The CA Single Sign-on WSS Agent Login Module then determines whether an authenticated user is allowed to access a protected WebLogic resource, based on associated CA Single Sign-On Web Services Security authorization policies.

## Installation Location References

In this document:

- *WSS_Home* refers to the location where CA Single Sign-On Web Services Security is installed.
- *WLS_HOME* refers to the installed location of the WebLogic Server.

# WSS Agent for WebLogic Install Preparation

**Contents**



## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.
2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network (http://www.oracle.com/technetwork/java/index.html).

## Software Requirements

Before installing the WSS Agent for Oracle WebLogic, install the following software:

**Note:** Be sure to install the prerequisite software in the correct order.

- A supported version of Oracle WebLogic Server and any cumulative fixes for this application server. For WebLogic hardware and software requirements, see the WebLogic documentation.
- A supported Java Runtime Environment (JRE).
- A Policy Server

**Note:** The Policy Server can be installed on a different system than the WebLogic Server.

For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

## Installation Checklist

Before you install the WSS Agent for Oracle WebLogic on the WebLogic Server, complete the steps in the following table. To ensure proper configuration, follow the steps in order. You can place a check in the first column as you complete each step.

| Completed? | Steps | For information, see... |
|---|---|---|
| | Install and configure the Policy Server. | Installing: Policy Server (see page 27) |
| | Install the Oracle WebLogic Server. | The Oracle WebLogic Server Documentation |
| | Configure the Policy Server for the WSS Agent for Oracle WebLogic. | Preconfigure Policy Objects for WSS Agents |
| | Install the WSS Agent on the Oracle WebLogic Server.<br>**Note**: For WebLogic clusters, install the WSS Agent on each node in the cluster. | Install the WSS Agent for WebLogic on a Windows System (see page 787)<br>or<br>2016-09-27_13-02-04_Install the WSS Agent for WebLogic on a UNIX System (see page 784) |

## Preconfigure Policy Objects for WSS Agents

This section describes how to preconfigure policy objects for WSS Agents on the Policy Server.

## Policy Object Preconfiguration Overview

Before you install any WSS Agent, the Policy Server must be installed and be able to communicate with the system where you plan to install the WSS Agent. Additionally, you must configure the Policy Server with the following:

- **An administrator that has the right to register trusted hosts**
  A trusted host is a client computer where one or more WSS Agents are installed. The term trusted host refers to the physical system. There must be an administrator with the privilege to register trusted hosts with the Policy Server.

- **Agent object/Agent identity**
  An Agent object creates an Agent identity by assigning the Agent a name. You define an Agent identity from the Agents object in the Administrative UI. You assign the Agent identity a name and specify the Agent type as a Web Agent.
  The name you assign for the Agent is the same name you specify in the DefaultAgentName parameter for the Agent Configuration Object that you must also define to centrally manage an Agent.

- **Host Configuration Object**
  This object defines the communication between the trusted host and the Policy Server after the initial connection between the two is made.
  A trusted host is a client computer where one or more WSS Agents can be installed. The term trusted host refers to the physical system, in this case the application server host.
  Do not confuse this object with the trusted host's configuration file, SmHost.conf, which is installed at the trusted host after a successful host registration. The settings in the SmHost.conf file enable the host to connect to a Policy Server for the first connection only. Subsequent connections are governed by the Host Configuration Object.

- **Agent Configuration Object**
  This object includes the parameters that define the Agent configuration. There are a few required parameters you must set for basic operation.
  The Agent Configuration Object must include a value for the DefaultAgentName parameter. This entry should match an entry you defined in the Agent object.

## Preconfigure the Policy Objects

The following is an overview of the configuration procedures you must perform on the Policy Server prior to installing the Agent software:

1. Duplicate or create a new Host Configuration Object, which holds initialization parameters for a Trusted Host. (If upgrading from an earlier Agent install, you can use the existing Host Configuration object).
   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

2. As necessary, add or edit parameters in the Host Configuration Object that you just created.

3. Create an Agent identity for the WSS Agent. You must select **Web Agent** as the Agent type for the WSS Agent.

4. Duplicate an existing or create a new Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

5. In the Agent Configuration Object you just created, ensure that the DefaultAgentName parameter is set to specify the Agent identity defined in Step 3.

# Install the WSS Agent for WebLogic on a Windows System

This section contains the following topics:



## Prepare the Java Environment on Windows

**Contents**



### Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.

2. Double-click System.

3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

### Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

***jre_home***

Defines the location of your Java Runtime Environment installation.

**Note:** If WebLogic is configured to use its own JRE then patch the JRE used by WebLogic to support unlimited key strength in the Java Cryptography Extension (JCE) package.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

   *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.
3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
```

```
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Run the Installer to Install a WSS Agent

Install the WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the installation material.
3. Double-click ca-sm-wss-*sm_version*-win32.exe.

   - ***sm_version***
     Specifies the release number and, if applicable, cumulative release number. The version does not include a cumulative release number in the file name.

   The CA Single Sign-On Web Services Security installation wizard starts.

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

4. Use gathered system and component information to install the WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agents for Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for Oracle WebLogic**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):
     - WebLogic 9.2: C:\bea\weblogic92
     - WebLogic 10.3 C:\bea\wlserver_10.3
     - WebLogic 11G: C:\Oracle\Middleware\wlserver_10.3
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system DLLs are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location.

6. On the CA Single Sign-On Web Services Security Configuration screen, click one of the following options and click Next:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

   If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

7. Click Done.
   If you selected the option to configure WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you installed a WSS Agent or Agents and did not select the option to configure WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The agent cannotcommunicate with the Policy Server until the trusted host is registered.

## Configure the WSS Agent for WebLogic and Register a Trusted Host on Windows

**Contents**

Configure the WSS Agent and register the system that hosts it as a trusted host using the CA Single Sign-On Web Services Security Configuration Wizard.

## Gather Information Required for WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed: Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1) Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like: policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

- **FIPS Compatibility Mode (Default)**
  Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

- **FIPS Only Mode**
  Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Run the CA Single Sign-on WSS Agent Configuration Wizard on Windows

You can configure your WSS Agent and register a trusted host immediately after installing the WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a WSS Agent on your system.

**Follow these steps:**

1. Open the following directory on your web server:
   *WSS_Home*\install_config_info

   - ***WSS_Home***
     Specifies the path to where CA Single Sign-On Web Services Security is installed.
     **Default:** C:\Program Files\CA\Web Services Security

2. Right-click ca-pep-config.exe, and then select Run as administrator.

   **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

   The WSS Agent Configuration Wizard starts.

3. Use gathered system and component information to configure the WSS Agent and register the host.

> **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*\config. You can modify this file.

- ***agent_home***
  Is the installed location of the WSS Agent.

## Modify the SmHost.conf File (Windows)

WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *agent_home*\config directory.
2. Open the SmHost.conf file in a text editor.
3. Enter new values for the any of the following settings that you want to change:

> **Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

- **hostconfigobject**

  Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

  If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

  **Example:** hostconfigobject="*host_configuration_object*"

- **policyserver**

  Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

"*IP_address*, *port*,*port*,*port*"

The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

**Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

**Default:***IP_address*, 44441,44442,44443

**Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

**Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

  Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

  **Default:** 60

  **Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool (Windows)

When you install a CA Single Sign-on WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.

- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home* \bin directory when you install a WSS Agent.

- ***agent_home***
  Is the installed location of the WSS Agent.

**To re-register a trusted host using the registration tool**

1. Open a command prompt window.
2. Enter the smreghost command using the following required arguments:

```
smreghost -i policy_server_IP_address:[port]
-u administrator_username -p Administrator_password
-hn hostname_for_registration -hc host_configuration_ object
```

**Note:** If the "-p Administrator_password" argument is not specified in the smreghost command, you are prompted to specify the password.

**Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

**-i *policy_server_IP_ address:port***

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

**Default:** (ports) 44441,44442,44443

**Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555

**Example**: (IPv4 default ports) -i 127.0.0.1

**Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

**Example:** (IPv6 default ports) -i [2001:DB8::/32]**-u *administrator_username***

Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

  Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**
  Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

  COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

  If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

  **Default:** COMPAT

  **Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

## Register Multiple Trusted Hosts on One System (Windows)

You typically register only one trusted host for each machine where web servers and agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each CA Single Sign-On client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

  **Note:** If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads: "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

  **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

## (Optional) Install a WSS Agent Using the Unattended Installer

After you have installed one or more WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home*\install_config_info***WSS_Home***

Specifies the path to where CA Single Sign-On Web Services Security is installed.

**Default:** C:\Program Files\CA\Web Services Security

**To run the installer in the unattended installation mode**

1. From a system where CA Single Sign-On Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.

2. Copy the WSS Agent installer file (ca-sm-wss-<SVMVER>-*cr*-win32.exe) into the same local directory as the ca-wss-installer.properties file.

   - ***cr***
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

3. Open a console window and navigate to the location where you copied the files.

4. Run the following command:

   ```
   ca-sm-wss-version-cr-win32.exe -f ca-wss-installer.properties -i silent
   ```

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

   The -i silent setting instructs the installer to run in the unattended installation mode.

   **Note:** If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   Example:

   ```
   ca-sm-wss-version-cr-win32.exe -f "C:
   \Program Files\CA\Web Services Security\install_config_info\ca-wss-installer.
   properties" -i silent
   ```

   An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-On Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log

  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security

  - ***install-date-and-time***
    Specifies the date and time that the CA Single Sign-on WSS Agent was installed.

- The Agent cannotcommunicate properly with the Policy Server until the trusted host is registered.
- To stop the installation manually, type Ctrl+C.

# Install the WSS Agent for WebLogic on a UNIX System

This section contains the following topics:



## Prepare the Java Environment on UNIX

**Contents**



### Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

**To set the JRE in the PATH variable**

1. Open a Command Window.
2. Run the following commands:

   ```
   PATH=$PATH:JRE
   export PATH
   ```

   *JRE* defines the location of your Java Runtime Environment bin directory.

### Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

***jre_home***

Defines the location of your Java Runtime Environment installation.

**Note:** If WebLogic is configured to use its own JRE then patch the JRE used by WebLogic to support unlimited key strength in the Java Cryptography Extension (JCE) package.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.
3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
```

```
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Run the Installer to Install a WSS Agent Using a GUI

Install the WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  `chmod +x ca-sm-wss-`*`sm_version`*`-`*`unix_version`*`.bin`

  - ***sm_version***
    Specifies the release and, if applicable, cumulative release number. The base version does not include a cumulative release number in the file name.

  - ***unix_version***
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**Follow these steps:**

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.
3. Enter the following command:

   `./ca-sm-wss-<SVMVER>-cr-unix_version.bin`

   The CA Single Sign-On Web Services Security installer starts.
4. Use gathered system and component information to install the WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agents for Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for Oracle WebLogic**.

   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

   - When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):

     - WebLogic 9.2: /bea/weblogic92

     - WebLogic 10.3 /bea/wlserver_10.3

- WebLogic 11G: /Oracle/Middleware/wlserver_10.3

- If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

- Do not use space characters in the WSS Agent install path. For example, "/CA Technologies /agent" will result in install failure.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location. Afterward, the CA CA Single Sign-On Web Services Security Configuration screen is displayed.

6. Select one of the following options:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.

   - No. I will configure CA Single Sign-On Web Services Security Agents later.

7. Click Done.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.

  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.

  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.

- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Run the Installer to Install a WSS Agent Using a UNIX Console

Install the WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  `chmod +x ca-sm-wss-sm_version-cr-unix_version.bin`

  - ***sm_version***
    Specifies the version and, where applicable, the cumulative release number. The base 12.52 SP1 release does not include a cumulative release number.

  - ***unix_version***
    Specifies the UNIX version: **sol** or **linux**..

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**To install the WSS Agent**

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.
3. Enter the following command:

   `./ca-sm-wss-sm_version-unix_version.bin -i console`

   The CA Single Sign-On Web Services Security installer starts.
4. Use gathered system and component information to install the WSS Agent. Consider the following as you make your selections:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agents for Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for Oracle WebLogic**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):
     - WebLogic 9.2: /bea/weblogic92
     - WebLogic 10.3 /bea/wlserver_10.3
     - WebLogic 11G: /Oracle/Middleware/wlserver_10.3

- When prompted for the location where WebLogic is installed, enter the correct location for your version of WebLogic.
- Do not use space characters in the WSS Agent install path. For example, "/CA Technologies /agent" will result in install failure.

5. Review the information presented on the Pre-Installation Summary page, then proceed.

   Note: If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location. Afterward, the CA Single Sign-On Web Services Security Configuration screen is displayed.

6. Select one of the following options:
   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

7. Hit Enter.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Configure the WSS Agent for WebLogic and Register a Trusted Host on UNIX

**Contents**

Configure a WSS Agent and register the system that hosts it as a trusted host using the CA Single Sign-On Web Services Security Configuration Wizard.

## Gather Information Required for WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Run the CA Single Sign-on WSS Agent Configuration Program on UNIX or Linux Systems

You can configure your WSS Agents and register a trusted host immediately after installing the WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a WSS Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.
- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

**To configure Agents and register a trusted host**

1. If necessary, start the Configuration Wizard as follows:
   a. Open a console window.
   b. Navigate to *agent_home*/install_config_info, where *agent_home* is the installed location of the WSS Agent.

c. Enter one of the following commands:
GUI Mode: ./ca-pep-config.bin
Console Mode: ./ca-pep-config.bin -i console

The Configuration Wizard starts.

2. Use gathered system and component information to configure the WSS Agent and register the host.

> **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*/config. You can modify this file.

- ***agent_home***
  Is the installed location of the WSS Agent

## Modify the SmHost.conf File

WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *agent_home*/config directory.

   - ***agent_home***
     Is the installed location of the WSS Agent.

2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

> **Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

   - **hostconfigobject**

Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

**Example:** hostconfigobject="*host_configuration_object*"

- **policyserver**

Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

"*IP_address*, *port*,*port*,*port*"

The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

**Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

**Default:***IP_address*, 44441,44442,44443

**Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

**Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

**Default:** 60

**Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool (UNIX)

When you install a CA Single Sign-on WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home* /bin directory when you install a WSS Agent.

- ***agent_home***

  Is the installed location of the CA Single Sign-on WSS Agent.

**To re-register a trusted host using the registration tool**

1. Open a command prompt window.
2. Ensure that the library path environment variable contains the path to the agent bin directory.
3. Enter the following two commands:
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:*agent_home*/bin
   export LD_LIBRARY_PATH
   For example, for a WSS Agent for WebLogic, enter the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/SOA Security Manager/wlsagent/bin
   export LD_LIBRARY_PATH
   ```

4. Enter the smreghost command using the following required arguments:

   ```
   smreghost -i policy_server_IP_address:[port]
   -u administrator_username -p Administrator_password
   -hn hostname_for_registration -hc host_configuration_ object
   ```

**Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

**-i *policy_server_IP_ address:port***

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

**Default:** (ports) 44441,44442,44443

**Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555

**Example**: (IPv4 default ports) -i 127.0.0.1

**Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

**Example:** (IPv6 default ports) -i [2001:DB8::/32]**-u *administrator_username***

Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

  Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**
  Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

  COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

  If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

  **Default:** COMPAT

  **Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

## Install a WSS Agent for WebLogic Using the Unattended Installer

After you have installed one or more WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home/*install_config_info

**To run the installer in the unattended installation mode**

1. From a system where CA Single Sign-On Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.

2. Copy the WSS Agent installer file (ca-sm-wss-*version-unix_version*) into the same local directory as the ca-wss-installer.properties file.

   - ***sm_version***
     Specifies the release number and, if applicable, cumulative release number. The base version does not include a cumulative release number in the file name.

   - ***unix_version***
     Specifies the UNIX version: **sol** or **linux**.

3. Open a console window and navigate to the location where you copied the files.

4. Run the following command:

   ```
   ./ca-sm-wss-sm_version-unix_version -f ca-wss-installer.properties -i silent
   ```

   The -i silent setting instructs the installer to run in the unattended installation mode.

   > **Note:** If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   Example:

   ```
   ./ca-sm-wss-sm_version-unix_version -f ~/CA/Web_Services_Security
   /install_config_info/ca-wss-installer.properties" -i silent
   ```

An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-On Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.
- To stop the installation manually, type Ctrl+C.

# Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time* _InstallLog.log file located in *WSS_Home*\install_config_info.

- ***date-time***
  Specifies the date and time of the CA Single Sign-On Web Services Security installation.

# Uninstall a WSS Agent for WebLogic

To uninstall a CA Single Sign-on WSS Agent, run the CA Single Sign-On Web Services Security uninstall wizard.

**Follow these steps:**

1. Navigate to the *WSS_HOME*\install_config_info (Windows) or *WSS_HOME*/install_config_info (UNIX) directory and run the CA Single Sign-On Web Services Security uninstall wizard to remove CA Single Sign-On Web Services Security agents:
   - Windows: soa-uninstall.cmd
   - UNIX: soa-uninstall.sh
   - ***WSS_HOME***
     Specifies the CA Single Sign-On Web Services Security installation location.

**Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.

3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.
   The uninstall wizard removes all selected CA Single Sign-On Web Services Security components.

4. Restart the server.

# WSS Agent Configuration Settings

**Contents**



## How to Configure the WSS Agent

To configure the WSS Agent, you must specify the following:

- Host Configuration Object (one for each host server)
- Agent Configuration Object (one for each WSS Agent)
- Agent identity (one for each WSS Agent)

**Follow these steps:**

1. On the Policy Server:

   a. Duplicate or create a Host Configuration Object, which holds initialization parameters for a Trusted Host.
   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

   b. As necessary, add or edit parameters in the Host Configuration Object that you just created.

c. Duplicate or create an Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

d. Add or edit required Agent parameters in the Agent Configuration Object.
The configuration object must include the DefaultAgentName or AgentName parameter to specify the Agent identity.

e. Create an Agent identity for the WSS Agent. You must select *Web Agent* as the Agent type for a WSS Agent.

2. On the system where the WSS Agent is installed:

a. Run the Agent Configuration Wizard, which registers the Trusted Host.

b. Enable the WSS Agent by setting the EnableWebAgent parameter in the Agent configuration file to Yes.

## WSS Agent for WebLogic Agent Configuration File

By default, the WSS Agent for WebLogic installation creates a single agent configuration file, JavaAgent.conf. The agent configuration file is located in the *WSS_Home*/config directory.

- ***WSS_Home***
  Specifies the location where the WSS Agent is installed.

Each agent configuration file is created with the following required default configuration parameters /values:

| Parameter | Description |
|---|---|
| DefaultAgentName | The agent identity the Policy Server uses to associate policies with the WSS Agent.<br>The default value is "SoaAgent". Do not change this value. |
| EnableAgent | Specifies whether the WSS Agent is enabled. Possible values are Yes and No.<br>Default value is Yes. |
| AgentConfigObject | The Agent Configuration Object specified during installation. |
| SmHostFile | Path to the local Host Configuration File. Path can be specified in absolute terms or relative to *WSS_HOME.*<br>**Note:** On Windows, specify paths using double backslashes ("\\") rather than single backslash ("\") to separate directories. On UNIX, use standard single slash ("/") separators.<br>Example values:<br>(Windows) C:\\Program Files\\CA\\Web Services Security\\wlsagent\\config\\SmHost.conf<br>(UNIX) /config/SmHost.conf |
| ServerName | A string that will be used in the WSS Agent log to identify the WebLogic Server.<br>Default value is "SOAWLS92". |
| appserverjaasloginhandler | Specifies the Application Server-specific WSS Agent handler class for WebLogic<br>Default value is "com.ca.soa.agent.appserver.jaas.wls.WlsLoginHandler". Do not change this value. |

You need only edit the preconfigured values if the location of the Host Configuration File changes or you want to refer to a different Agent Configuration Object. If you use local configuration, you can add other Agent configuration parameters to these preconfigured values.

**Note:** Parameters that are held in the Agent configuration file are static. If you change these settings while the WebLogic server is running, the WSS Agent does not pick up the change until WebLogic is restarted.

The JavaAgent.conf file also contains a list of WSS Agent plugin classes; you do not need to alter this information.

**Note**: Leading and trailing whitespace in JavaAgent.conf value definitions is ignored. To include leading or trailing whitespace, quote the value (with either single or double quotes). Embedded, escaped quotes are unescaped during processing.

**Sample JavaAgent.conf (Windows)**

```
# Java Agent Configuration File
#
# This file contains bootstrap information required by
# the SiteMinder Java Agent
#
#
# Configuration for agent testagent
#
defaultagentname=soaagent
enablewebagent=yes
agentconfigobject=soaagentconfig
servername=wsdell110.systemtest.local
smhostfile=C:\\SOASecurityManager\\wlsagent\\config\\SmHost.conf
appserverjaasloginhandler=com.ca.soa.agent.appserver.jaas.wls.WLSLoginHandler
appserverjmshandler=com.ca.soa.agent.appserver.jaxrpc.jms.wls.WLSJMSMessageHandler
# Configure plugins for the agent testagent
transport_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.jaxrpcplugin.pluginconfig.JaxRpcPluginConfig, com.ca.soa.agent.jmsplugin.
pluginconfig.JMSPluginConfig
msg_body_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
credential_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
variable_resolver_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
# <EOF>
```

## Agent Configuration Object

An Agent Configuration Object is a CA Single Sign-On policy object that holds Agent parameters for an Agent when using central agent configuration.

**Note:** Parameters held in an Agent Configuration Object are dynamic; if you change these settings while the WebLogic server is running, the WSS Agent will pick up the change.

## WSS Agent Configuration Parameters

The following table contains a complete list of all Agent configuration parameters supported by WSS Agents for Application Servers.

Unless otherwise noted,you can define parameters in either the Agent Configuration Object or the Agent configuration file depending upon how you decide to configure the WSS Agent.

| Parameter Name | Value | Description |
|---|---|---|
| AcceptTPCookie | YES or NO | (Optional) If set to yes, configures the WSS Agent to assert identities from third-party CA Single Sign-on session cookies (that is, session cookies generated by custom Agents created using the CA Single Sign-on and CA Single Sign-on WSS SDKs. |
| AllowLocalConfig (Applies only in the Agent Configuration Object) | YES or NO | If set to yes, parameters set locally in the Agent configuration file take precedence over parameters in the Agent Configuration Object.<br>Default is NO. |
| AuthCacheSize | Number | (Optional) Size of the authentication cache for the WSS Agent (in number of entries). For example:<br>`authcachesize="1000"`<br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| AzCacheSize | Number | (Optional) Size of the authorization cache (in number of entries) for the WSS Agent. For example:<br>`authcachesize="1000"`<br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| CacheTimeout | Number | (Optional) Number of seconds before cache times out. For example:<br>`cachetimeout="1000"`<br>Default is 600 (10 minutes). |
| ConfigObject (Applies only in Agent configuration file) | String | The name of the Agent Configuration Object associated with the WSS Agent.<br>No default value. |
| CookieDomain | String | (Optional) Name of the cookie domain. For example:<br>`cookiedomain="ca.com"`<br>No default value.<br>For more information, see the cookiedomainscope parameter. |

| Parameter Name | Value | Description |
|---|---|---|
| CookieDomainScope | Number | (Optional) Further defines the cookie domain for assertion of CA Single Sign-on session cookies by the WSS Agent. The scope determines the number of sections, separated by periods, that make up the domain name. A domain always begins with a period (.) character. For example:<br>`cookiedomainscope="2"`<br><br>Default is 0, which takes the domain name specified in the cookiedomain parameter. |
| DefaultAgentName (Applies only in the Agent Configuration Object) | String | The agent identity the Policy Server will use to associate policies with the WSS Agent.<br>Default is "SoaAgent"; this value should not changed. |
| EnableWebAgent (Applies only in Agent configuration file) | YES or NO | Enables or disables the WSS Agent. When set to 'yes', the WSS Agent will protect resources using the Policies configured in the Policy Server for the configured agent identity.<br>Default is Yes. |
| LogOffUri | String | (Optional) The URI of a custom HTTP file that will perform a full log off (removing the session cookie from a user's browser). A fully qualified URI is not required. For example, LogOffUri could be set to: /Web pages/logoff.html<br>No default value. |
| PsPollInterval | Number | (Optional) The frequency with which the WSS Agent polls the Policy Server to retrieve information about policy changes.<br>Default is 30 seconds. |
| ResourceCacheSize | Number | (Optional) Size (in number of entries) of the cache for resource protection decisions. For example:<br>`resourcecachesize="1000"`<br><br>Default is 2000.<br>To flush this cache, use the Administrative UI. |
| SAMLSessionTicketLogoffi | YES or NO | (Optional) Determines whether the WSS Agent should attempt to log off session tickets in SAML assertions.<br>Default is Yes. |
| ServerName (Applies only in Agent configuration file.) | String | A string to be used in the WSS Agent log to identify the target application server. |
| SessionGracePeriod | Number | (Optional) Grace period (in seconds) between the regeneration of session tokens.<br>Default is 30 |
| | String | Path to the local Host Configuration File (typically *WSS_Home*\conf\SmHost.conf).<br>No default value. |

| Parameter Name | Value | Description |
|---|---|---|
| SmHostFile (Applies only in Agent configuration file) | | |
| XMLAgentSoapFaultDetails | YES or NO | (Optional) Determines whether or not the WSS Agent should insert the authentication/authorization rejection reason (if provided by the Policy Server) into the SOAP fault response sent to the web service consumer.<br>Default is No. |
| XMLSDKAcceptSMSessionCookie | YES or NO | (Optional) Determines whether or not the WSS Agent accepts a CA Single Sign-on session cookie to authenticate a client.<br>Default is No.<br>If set to Yes, the WSS Agent uses information in a session cookie sent as an HTTP header in the request as a means of authenticating the client.<br>If set to No, session cookies are ignored and the WSS Agent requests credentials required by the configured authentication scheme. |
| XMLSDKMimeTypes | String | (Optional) A comma-delimited list of MIME types that the WSS Agent will accept for processing by CA Single Sign-On Web Services Security. All POSTed requests having one of the listed MIME types are processed. Examples:<br>text/xml<br>application/octet-stream<br>text/xml,multipart/related<br>If you do not add this parameter to the Agent Configuration Object, the WSS Agent defaults to accepting text/xml and application/soap+xml MIME types. |

## Configure the Username and Password Digest Token Age Restriction

By default, the WS-Security authentication scheme imposes a 60-minute restriction on the age of Username and Password Digest Tokens to protect against replay attacks.

To configure a different value for the token age restriction for a WSS Agent for Application Servers, set the WS_UT_CREATION_EXPIRATION_MINUTES parameter in the XmlToolkit.properties file for that agent.

**Follow these steps:**

1. Navigate to *WSS_Home**wlsagent*\config.
2. Open XmlToolkit.properties in a text editor.
3. Uncomment and modify the WS_UT_CREATION_EXPIRATION_MINUTES parameter line to configure a different value for the token age restriction:

   ```
   WS_UT_CREATION_EXPIRATION_MINUTES=token_age_limit
   ```

   - ***token_age_limit***
     Specifies the token age limit restriction in minutes.
4. Save and close the XmlToolkit.properties file.

5. Restart the CA Single Sign-on WSS Agent.

# Set the WebLogic Environment for the WSS Agent

**Contents**



## WebLogic Environment Setting Locations

You configure WSS Agent-related environment settings in one of the following locations depending on your environment:

- The WebLogic start script for both managed and standalone servers (startWebLogic.cmd on Windows; startWebLogic.sh on UNIX)
  **Note:** The startWebLogic.cmd (Windows) or startWebLogic.sh (Unix) script that contains the environment configuration is placed in the bin folder of a created domain.

- If using the Node Manager to control Managed Servers, in the Server Start configuration page in the WebLogic Adminstration Console.
  For details regarding the Server Start configuration page, see the WebLogic Online Documentation.

## Set the WebLogic Environment on Windows

Before the WSS Agent can operate with the WebLogic Application Server on Windows, you must configure WSS Agent-related environment settings.

**Follow these steps:**

1. Define a SOA_HOME variable as follows:

   ```
   set SOA_HOME=WSS_Home\wlsagent
   ```

   ***WSS_Home***
   Specifies the path to where CA Single Sign-On Web Services Security is installed.

2. Define a SMSOA_CLASSPATH variable as follows:

   ```
   set SMSOA_CLASSPATH=%SOA_HOME%\config;
   %SOA_HOME%\lib\smagentapi.jar;
   %SOA_HOME%\lib\thirdparty\cryptojFIPS.jar;
   %SOA_HOME%\lib\soaagent-proxy.jar;
   %SOA_HOME%\lib\thirdparty\xalan.jar
   ```

3. Add %SMSOA_CLASSPATH% to the beginning of the CLASSPATH variable definition. The modified CLASSPATH variable should resemble the following:

   ```
   set CLASSPATH=%SMSOA_CLASSPATH%;%CLASSPATH%
   ```

4. Define the SM_JAVA_OPTIONS variable as follows:

```
set SM_JAVA_OPTIONS=-DJAVA_AGENT_ROOT=%SOA_HOME%
-Dlog.log-config-properties=%SOA_HOME%\config\log-config.properties
-Djava.security.auth.login.config==%SOA_HOME%\config\soa_jaas.config
-Djavax.xml.soap.SOAPFactory=weblogic.xml.saaj.SOAPFactoryImpl
-Djavax.xml.soap.MessageFactory=weblogic.xml.saaj.MessageFactoryImpl
```

5. Add %SM_JAVA_OPTIONS% to the execution entry. The modified execution entry should resemble the following:

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
%SM_JAVA_OPTIONS% -Dweblogic.Name=%SERVER_NAME%
-Djava.security.policy=%WL_HOME%\server\lib\weblogic.policy %PROXY_SETTINGS% %
SERVER_CLASS%
```

6. Save your changes.

7. Restart the WebLogic Application Server for changes to take effect.

## Set the WebLogic Environment on UNIX

Before the WSS Agent can operate with the WebLogic Application Server on UNIX, you must configure WSS Agent-related environment settings.

**Follow these steps:**

1. Define a SOA_HOME variable as follows:

```
SOA_HOME=WSS_Home/wlsagent
```

2. Define the SMSOA_CLASSPATH as follows:

```
SMSOA_CLASSPATH=${SOA_HOME}/config:
${SOA_HOME}/lib/smagentapi.jar:
${SOA_HOME}/lib/thirdparty/cryptojFIPS.jar:
${SOA_HOME}/lib/soaagent-proxy.jar:
${SOA_HOME}/lib/thirdparty/xalan.jar
export SMSOA_CLASSPATH
```

3. Add ${SMSOA_CLASSPATH} to the beginning of the CLASSPATH definition. The modified CLASSPATH variable should resemble the following:

```
CLASSPATH=${SMSOA_CLASSPATH}${CLASSPATHSEP}${CLASSPATH}
export CLASSPATH
```

4. Define the SM_JAVA_OPTIONS variable as follows:

```
SM_JAVA_OPTIONS="-DJAVA_AGENT_ROOT=${SOA_HOME}
-Dlog.log-config-properties=${SOA_HOME}/config/log-config.properties
-Djava.security.auth.login.config==${SOA_HOME}/config/soa_jaas.config
-Djavax.xml.soap.SOAPFactory=weblogic.xml.saaj.SOAPFactoryImpl
-Djavax.xml.soap.MessageFactory=weblogic.xml.saaj.MessageFactoryImpl"
```

5. Add ${SM_JAVA_OPTIONS} to the execution entry. The modified execution entry should resemble the following:

```
"${JAVA_HOME}/bin
/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS} ${SM_JAVA_OPTIONS} -Dweblogic.
Name=${SERVER_NAME} -Djava.security.policy=${WL_HOME}/server/lib/weblogic.
policy ${PROXY_SETTINGS} ${SERVER_CLASS}"
```

6. Save your changes.

7. Restart the WebLogic Application Server for changes to take effect.

# WSS Agent for Oracle WebLogic Logging

**Contents**



## log4j

The WSS Agent logger for application servers is implemented using Apache's log4j. For more information, see http://logging.apache.org/log4j/docs/.

## Log Files

Two log files provide important information about the WSS Agent:

- WSS Agent log file -- Logs WSS Agent error and processing messages.
- WSS Agent XML message processing log file -- Logs messages information relating specifically to the WSS Agent processing of XML messages

### WSS Agent Log

The WSS Agent for Oracle WebLogic writes information about its standard operations and performance to the WSS Agent log.

By default, WSS Agent logging is enabled and written to the XmlAgent.log file in:

- Windows -- *WLS_HOME*\user_projects\domain\defaultdomain\soa-log
- UNIX -- *WLS_HOME*/user_projects/domain/defaultdomain/soa-log

You can change WSS Agent logging parameters by editing the log-config.properties file located in:

- Windows -- *WSS_Home*\wlsagent\config\
- UNIX -- *WSS_Home*/wlsagent/config/

**Note:** These are the default values; the logging configuration file name and location can be changed by editing the log-config-properties JVM system property.

### WSS Agent XML Message Processing Logging

In addition to its standard logging functionality, WSS Agents for Application Servers also logs information relating specifically to its processing of XML messages. Like the WSS Agent log, the XML message processing log is also implemented using Apache's *log4j* standard.

**Note:** WSS Agent XML message processing logging does not start until an XML message that needs to be processed is received.

By default, WSS Agent XML message processing logging is enabled and written to the soasm_agent.log file in:

- Windows -- *WSS_Home*\wlsagent\bin\
- UNIX -- *WSS_Home*/wlsagent/bin/

You can change WSS Agent XML message processing logging parameters by editing the log.config file, which can be found in:

- Windows -- *WSS_Home*\wlsagent\config\
- UNIX -- *WSS_Home*/wlsagent/config/

## Change the WSS Agent Log File Name

To change pathname of the WSS Agent log file, edit the log.logfile-pattern parameter. Possible values are valid pathnames. If you specify a relative value, the path is set relative to the JAVA_AGENT_ROOT JVM system property.

Default value: "log\XmlAgent.log"

For example:

```
log.logfile-pattern=log\XmlAgent.log
```

## Append Messages to an Existing WSS Agent Log File

To add logging information to an existing WSS Agent log file instead of rewriting the entire file each time logging is invoked, add the log.logfile-append-on-reset parameter.

For example:

```
log.logfile-append-on-reset=YES
```

## Set the WSS Agent File Log Level

To change the WSS Agent log level, edit the log.logging-level parameter. Possible values are:

- DEBUG - Logs all; most verbose
- CONFIG - Configuration information
- INFO - Information
- WARNING - Warnings
- SEVERE - Errors only; least verbose

Default value: WARNING

For example:

```
log.logging-level=INFO
```

## Roll Over the WSS Agent Log File

To change file size limit at which the WSS Agent log should rollover, change the log.logfile-limit parameter. Rolling over a log file starts a new log file, preventing a single log file from becoming unmanageable. Possible values are numbers, representing kilobytes.

The default value is 1000.

For example:

```
log.logfile-limit=512
```

## Disable WSS Agent XML Message Processing Logging

To disable WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent config subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

## WSS Agent Log Configuration File Summary

The WSS Agent logging configuration file defines default WSS Agent logging settings.

Available configuration parameters are:

| Name | Description |
|---|---|
| log.logfile-append-on-reset | Add logging information to an existing log file instead of creating a new file each time logging is invoked.<br>Default value: no |
| log.logfile-pattern | Specifies the pathname (relative to *WSS_Home*) of the WSS Agent log file.<br>Default value: log/XmlAgent.log |

| Name | Description |
|---|---|
| log.logging-level | Defines the logging level. The levels are:<br>DEBUG - all logging, most verbose<br>CONFIG - configuration information<br>INFO - information<br>WARNING -warnings<br>SEVERE - errors<br>Default value: WARNING |
| log.logfile-limit | Specifies the size limit, in kilobytes Rollover a log file after it reaches the specified size.<br>Default value: 1,000KB |

**Note:** Once the WSS Agent connects to the Policy Server, corresponding logging settings found in the Agent Configuration Object override the values in log-config.properties.

# Finalize the WSS Agent for WebLogic Installation

**Contents**



## Prevent WebLogic 10 from Loading Incompatible Version of XML Security

By default, Weblogic Server 10 loads an older version of XML security (1.3.0) that is incompatible with the version used by the WSS Agent (1.4.1).

To prevent WebLogic 10 from loading the 1.3.0 XML security JAR, rename *WLS_HOME*\modules\com.bea.core.apache.xml.security_1.3.0.jar to some other name. For example, com.bea.core.apache.xml.security_1.3.0_backup.jar.

## Restart WebLogic

After completing WebLogic-side configuration of the WSS Agent, you must restart the WebLogic server.

## Configure Web Services to Invoke the WSS Agent JAX-RPC Handler

To protect a JAX-RPC web service using the WSS Agent, you must configure it to invoke the WSS Agent JAX-RPC Handler. To do this, you must add the WSS Agent JAX-RPC Handler class (com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandlerProxy) to the web service deployment descriptor in the webservices.xml file.

You can do this manually by editing the webservices.xml file for each JAX-RPC web service module. However, if your web services are implemented as JWS files and you have set up an Ant-based development environment, it is more efficient to update your web services to use handler chains.

## Manually Edit JAX-RPC Web Service Deployment Descriptors

To configure JAX-RPC web services not implemented as JWS files to invoke the WSS Agent, you must manually edit their deployment descriptors to add the WSS Agent JAX-RPC Handler.

**Follow these steps:**

1. Unpack the enterprise archive (EAR) containing one or more web services.
2. Examine the EAR to determine which of the modules within it contains a JAX-RPC web service. (A module that contains a JAX-RPC web service if it has a webservices.xml file in the META-INF folder for EJB endpoints, or the WEB-INF folder for servlet endpoints.)
3. For each module in the EAR identified as a JAX-RPC web service:
    a. Unpack the archive containing the module. (The archive will be a JAR file for EJB endpoints and a WAR file for servlet endpoints.)
    b. Find the webservices.xml file.
    c. For each port-component element found in the webservices.xml file, add a handler element:

```
<handler>
   <handler-name>WSS Agent Handler<handler-name>
   <handler-class>
      com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandlerProxy
   </handler-class>
</handler>
```

> **Note:** The WSS Agent JAX-RPC handler must always be invoked first; If other handler elements are already present or subsequently added to the webservices.xml file, the WSS Agent JAX-RPC Handler element must be placed before them.

4. Repackage the module into the appropriate archive type (JAR or WAR).
5. When all modules have been configured, repackage the EAR.
6. Install or update the enterprise application.

## Use Handler Chains

The most efficient way to configure services implemented as JWS files to invoke the WSS Agent is to define the WSS Agent JAX-RPC Handler class in a handler chain configuration file which can then be referenced from the JWS files of all web services in your enterprise that you need to protect.

**Note:** The following procedures assume that you have set up an Ant-based development environment and have a working build.xml file that includes a target for running the jwsc Ant task.

## Use a Handler Chain to Invoke the WSS Agent for HTTP Requests

To configure services implemented as JWS files to invoke the WSS Agent for requests received over HTTP transport, define the WSS Agent JAX-RPC Handler class in a handler chain configuration file.

**Follow these steps:**

1. Create a handler chain configuration file that defines a JAX-RPC handler chain. The chain can include as many handler classes as you require but must define the WSS Agent JAX-RPC Handler class first.
   Example HandlerConfig.xml:

```
<jwshc:handler-config xmlns:jwshc="http://www.bea.com/xml/ns/jws"
 xmlns:soap1="http://HandlerInfo.org/Server1"
 xmlns:soap2="http://HandlerInfo.org/Server2"
 xmlns="http://java.sun.com/xml/ns/j2ee" >
  <jwshc:handler-chain>
   <jwshc:handler-chain-name>HandlerChainName</jwshc:handler-chain-name>
   <jwshc:handler>
      <handler-name>handlerOne</handler-name>
      <handler-class>com.ca.soa.agent.appserver.jaxrpc.
XMLAgentJaxrpcHandlerProxy
      </handler-class>
    </jwshc:handler>
  </jwshc:handler-chain>
</jwshc:handler-config>
```

   - ***HandlerChainName***
     Specifies the name of handler chain.

2. Add the JWS annotation @HandlerChain(file="HandlerConfig.xml", name="HandlerChainName") to the web service JWS file.

3. Rebuild the JWS web service.

WebLogic server will invoke WSS Agent JAX-RPC handler.

**Note:** For more information on SOAP message handlers and handler chains, see the WebLogic documentation.

## Use a Handler Chain to Invoke the WSS Agent for JMS Requests

To configure services implemented as JWS files to invoke the WSS Agent for requests received over JMS transport, define the WSS Agent JAX-RPC Handler class in a handler chain configuration file.

**Follow these steps:**

1. Create a handler chain configuration file that defines a JAX-RPC handler chain. The chain can include as many handler classes as you require but must define the WSS Agent JAX-RPC Handler class first.
   Example HandlerConfig.xml:

```
<jwshc:handler-config xmlns:jwshc="http://www.bea.com/xml/ns/jws"
 xmlns:soap1="http://HandlerInfo.org/Server1"
```

```
xmlns:soap2="http://HandlerInfo.org/Server2"
xmlns="http://java.sun.com/xml/ns/j2ee" >
 <jwshc:handler-chain>
  <jwshc:handler-chain-name>HandlerChainName</jwshc:handler-chain-name>
  <jwshc:handler>
     <handler-name>handlerOne</handler-name>
     <handler-class>
       com.ca.soa.agent.appserver.jaxrpc.jms.XMLAgentJMSJaxrpcHandlerProxy
     </handler-class>
   </jwshc:handler>
 </jwshc:handler-chain>
</jwshc:handler-config>
```

- ***HandlerChainName***
  Specifies the name of handler chain.

2. Add the JWS annotation @HandlerChain(file="HandlerConfig.xml", name="HandlerChainName") to the web service JWS file.

3. Rebuild the JWS web service.

WebLogic server will invoke WSS Agent JAX-RPC handler for JMS requests.

**Note:** For more information on SOAP message handlers and handler chains, see the WebLogic documentation.

### Configure Policies for the WSS Agent

You create authentication and authorization policies to protect web service resources hosted on WebLogic from their associated WSDL files using the Administrative UI.

# Web Services Security Agent for IBM WebSphere

The following sections detail how to install and configure a WSS agent on IBM WebSphere.

# WSS Agent for IBM WebSphere Introduced

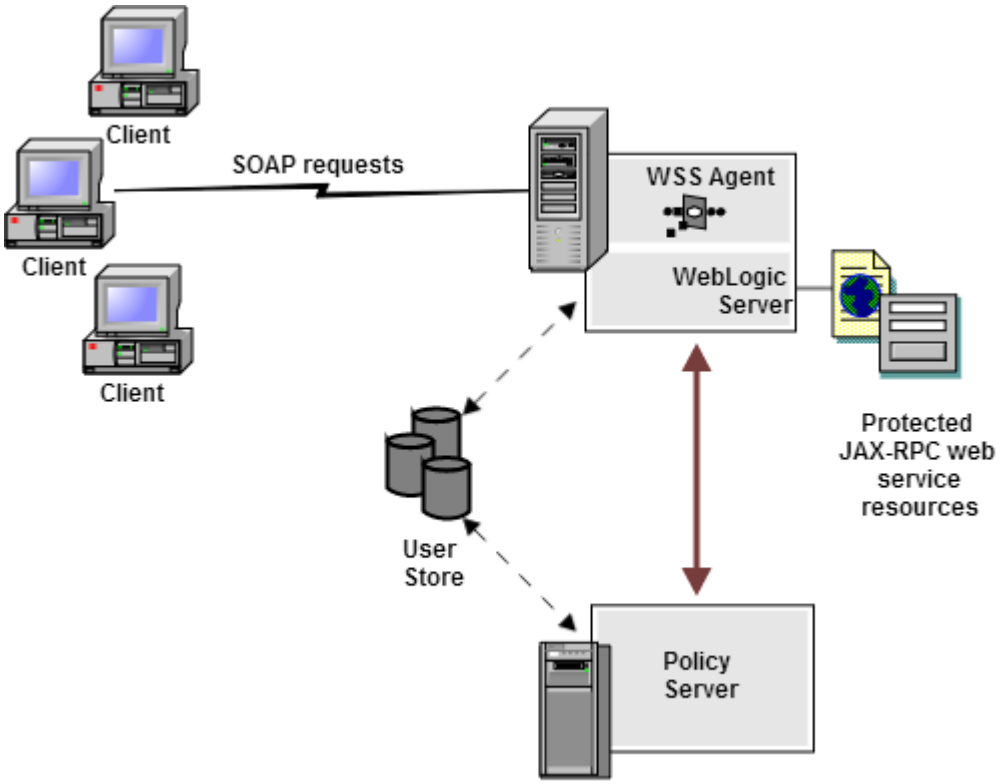
**Contents**

## CA Single Sign-on WSS Agent for IBM WebSphere Overview

The Web Services Security (WSS) Agent for IBM WebSphere resides in a WebSphere Application Server, enabling you to protect WebSphere-hosted JAX-RPC web service resources.

The WSS Agent for IBM WebSphere intercepts all SOAP messages sent over HTTP or HTTPS transport to JAX-RPC web services deployed on the Websphere Application Server. The WSS Agent then communicates with the Policy Server to authenticate and authorize the message sender and, upon successful authentication and authorization, passes the SOAP message on to the addressed web service.

A high-level overview of the WSS Agent for IBM WebSphere Server architecture is shown in the following figure:




The WSS Agent for IBM WebSphere provides the following features:

- CA Single Sign-On Web Services Security Integration with the J2EE platform
- Fine-grained access control of JAX-RPC web service resources
- Support for bi-directional CA Single Sign-On Web Services Security/CA Single Sign-On and WebSphere single sign-on (SSO)
- Support for WebSphere clustering

The WSS Agent additionally supports:

- J2EE RunAs identity
- Multi-byte character usernames
- User mapping to support environments in which WebSphere and CA Single Sign-On Web Services Security are not configured to use the same user store
- Centralized and dynamic agent configurations
- Caching of resource protection decisions and authentication and authorization decisions
- Logging
- Authorization auditing

## Required Background Information

This document assumes that you have the following technical knowledge:

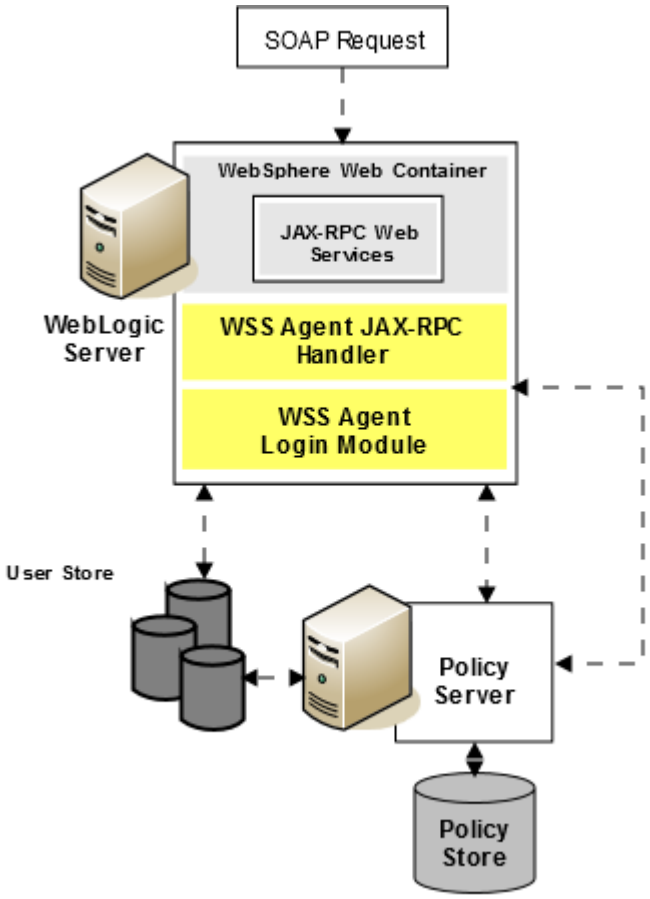
- An understanding of Java, J2EE standards, J2EE application servers, and multi-tier architecture
- An understanding of JAX-RPC web service implementations and JAX-RPC handlers
- Experience with the IBM WebSphere Application Server, its architecture and security infrastructure.
- Familiarity with Java Authentication and Authorization Server (JAAS) and WebSphere security-related topics
- Familiarity with CA Single Sign-On Web Services Security concepts, terms, and Policy Server configuration tasks

Additionally, to effectively plan your security infrastructure, you must be familiar with the web services that you plan to protect with CA Single Sign-On Web Services Security.

## WSS Agent for IBM WebSphere Components

The WSS Agent for IBM WebSphere consists of two modules that plug into WebSphere's security infrastructure.

- WSS Agent JAX-RPC Handler (see page )
- WSS Agent Login Module (see page )




## WSS Agent JAX-RPC Handler

The WSS Agent JAX-RPC Handler is a custom JAX-RPC Handler that, when added to the deployment descriptor of a JAX-RPC web service, intercepts SOAP message requests for JAX-RPC web services and diverts them to the WSS Agent Login Module for authentication and authorization decisions.

## WSS Agent Login Module

The WSS Agent Login Module is a JAAS Login Module that performs authentication and authorization for JAX-RPC web services protected by the WSS Agent for IBM WebSphere.

The WSS Agent Login Module authenticates credentials obtained from the following request types against associated user directories configured in CA Single Sign-On Web Services Security:

- SOAP requests intercepted by the WSS Agent JAX-RPC Handler .
- Requests for web service resources from users with pre-established CA Single Sign-On Web Services Security and CA Single Sign-on sessions (validating the session and obtaining user names from associated CA Single Sign-on session ticket cookies)
- System login (such as J2EE RunAs identity) requests.

If CA Single Sign-On Web Services Security authentication is successful, the WSS Agent Login Module populates a JAAS Subject with a CA Single Sign-On Web Services Security Principal that contains the username and associated CA Single Sign-On Web Services Security session data.

The WSS Agent Login Module then determines whether an authenticated user is allowed to access a protected WebSphere resource, based on associated CA Single Sign-On Web Services Security authorization policies.

## Recommended Reading List

To learn about the WebSphere Application Server and Java, see the following resources:

- IBM Redbooks Online
- IBM WebSphere Application Server Information Center
- Sun Microsystems, Inc., online documentation
  .

## Installation Location References

In this document:

- *WSS_HOME* refers to the location where CA Single Sign-On Web Services Security is installed.
- *WAS_HOME* refers to the installed location of the WebSphere Application Server.

# Prepare to Install a WSS Agent for IBM WebSphere

**Contents**



## Locate the Platform Support Matrix

Use the Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM) to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Go to the CA Support site (http://support.ca.com).
   The Welcome page displays.

2. In the top field, start typing the product name you are interested in then select the appropriate entry.
3. Mouse over the Knowledge Center option and click **Product Pages**.
4. On the right side of the page, Under Popular Links, click **Platform Support Matrices**.
5. At the top of the page, click **Platform Support Matricies** again to go directly to the correct table.
6. Select the PDF for the version you want.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Software Requirements

Before installing the CA Single Sign-on WSS Agent for IBM WebSphere, install the following software:

**Note:** Be sure to install the prerequisite software in the correct order.

- A supported version of IBM WebSphere Application Server and any cumulative fixes for this application server. For WebSphere hardware and software requirements, see the WebSphere documentation.
- CA Single Sign-On Policy Server

**Note:** The Policy Server can be installed on a different system than the WebSphere Application Server.

For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

## Installation Checklist

Before you install the CA Single Sign-on WSS Agent for IBM WebSphere on the WebSphere server, complete the steps in the following table. To ensure proper configuration, follow the steps in order. You can place a check in the first column as you complete each step.

| Completed? | Steps | For information, see... |
|---|---|---|
| | Install and configure the CA Single Sign-On Policy Server. | *Installing: Policy Server* |
| | Install the IBM WebSphere Application Server. | The IBM WebSphere Application Server Documentation |
| | Configure the Policy Server for the CA Single Sign-on WSS Agent for IBM WebSphere. | Preconfiguring Policy Objects for CA Single Sign-on WSS Agents |
| | Install the CA Single Sign-on WSS Agent on the WebSphere Application Server.<br>Note: For WebSphere clusters, install the CA Single Sign-on WSS Agent on each node in the cluster. | Install a CA Single Sign-on WSS Agent on a Windows System<br>or<br>Install a CA Single Sign-on WSS Agent on a UNIX System |

## Preconfigure Policy Objects for CA Single Sign-on WSS Agents

This section describes how to preconfigure policy objects for CA Single Sign-on WSS Agents on the Policy Server.

### Policy Object Preconfiguration Overview

Before you install any CA Single Sign-on WSS Agent, the CA Single Sign-On Web Services Security Policy Server must be installed and be able to communicate with the system where you plan to install the CA Single Sign-on WSS Agent. Additionally, you must configure the Policy Server with the following:

- **An administrator that has the right to register trusted hosts**
  A trusted host is a client computer where one or more CA Single Sign-on WSS Agents are installed. The term trusted host refers to the physical system. There must be an administrator with the privilege to register trusted hosts with the Policy Server.

- **Agent object/Agent identity**
  An Agent object creates an Agent identity by assigning the Agent a name. You define an Agent identity from the Agents object in the Administrative UI. You assign the Agent identity a name and specify the Agent type as a Web Agent.
  The name you assign for the Agent is the same name you specify in the DefaultAgentName parameter for the Agent Configuration Object that you must also define to centrally manage an Agent.

- **Host Configuration Object**
  This object defines the communication between the trusted host and the Policy Server after the initial connection between the two is made.
  A trusted host is a client computer where one or more CA Single Sign-on WSS Agents can be installed. The term trusted host refers to the physical system, in this case the application server host.
  Do not confuse this object with the trusted host's configuration file, SmHost.conf, which is installed at the trusted host after a successful host registration. The settings in the SmHost.conf file enable the host to connect to a Policy Server for the first connection only. Subsequent connections are governed by the Host Configuration Object.

- **Agent Configuration Object**
  This object includes the parameters that define the CA Single Sign-on Agent configuration. There are a few required parameters you must set for basic operation.
  The Agent Configuration Object must include a value for the DefaultAgentName parameter. This entry should match an entry you defined in the Agent object.

### Preconfiguring the Policy Objects

The following is an overview of the configuration procedures you must perform on the Policy Server prior to installing the Agent software:

1. Duplicate or create a new Host Configuration Object, which holds initialization parameters for a Trusted Host. (If upgrading from an earlier Agent install, you can use the existing Host Configuration object).
   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

2. As necessary, add or edit parameters in the Host Configuration Object that you just created.

3. Create an Agent identity for the CA Single Sign-on WSS Agent. You must select **Web Agent** as the Agent type for the CA Single Sign-on WSS Agent.

4. Duplicate an existing or create a new Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

5. In the Agent Configuration Object you just created, ensure that the DefaultAgentName parameter is set to specify the Agent identity defined in Step 3.

## Install a WSS Agent for WebSphere on Windows

This section contains the following topics:



### Prepare the Java Environment for the WSS Agent for WebSphere on Windows

**Contents**



### Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.
2. Double-click System.
3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

## Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package.

The WebSphere JRE is based on Sun's JRE on the Solaris platform; this patch is available at the Sun website. The patch for other platforms is available at IBM's website. See the IBM documentation for more details.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows
  *WAS_HOME*\java\jre\lib\security
- UNIX
  *WAS_HOME*/java/jre/lib/security

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:
   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.
2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.

3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## Install the WSS Agent for WebSphere on Windows

**Contents**



### Run the Installer to Install a CA Single Sign-on WSS Agent

Install the CA Single Sign-on WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to the installation material.
3. Double-click ca-sm-wss-12.52-*cr*-win32.exe.

   - ***cr***
     Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

   The CA Single Sign-On Web Services Security installation wizard starts.

> **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

4. Use gathered system and component information to install the CA Single Sign-on WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agentsfor Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for IBM WebSphere**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system DLLs are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The CA Single Sign-on WSS Agent files are copied to the specified location.

6. On the CA Single Sign-On Web Services Security Configuration screen, click one of the following options and click Next:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

   If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

7. Click Done.
   If you selected the option to configure CA Single Sign-on WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you installed a CA Single Sign-on WSS Agent or Agents and did not select the option to configure CA Single Sign-on WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log

  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security

- ***install-date-and-time***
  Specifies the date and time that the CA Single Sign-on WSS Agent was installed.

- The Agent cannotcommunicate properly with the Policy Server until the trusted host is registered.

## (Optional) Install a CA Single Sign-on WSS Agent Using the Unattended Installer

After you have installed one or more CA Single Sign-on WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall CA Single Sign-on WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home*\install_config_info

***WSS_Home***

Specifies the path to where CA Single Sign-On Web Services Security is installed.

**Default:** C:\Program Files\CA\Web Services Security

**To run the installer in the unattended installation mode**

1. From a system where CA Single Sign-On Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.

2. Copy the CA Single Sign-on WSS Agent installer file (ca-sm-wss-<SVMVER>-*cr*-win32.exe) into the same local directory as the ca-wss-installer.properties file.

   - ***cr***
     Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

3. Open a console window and navigate to the location where you copied the files.

4. Run the following command:

   ```
   ca-sm-wss-<SVMVER>-cr-win32.exe -f ca-wss-installer.properties -i silent
   ```

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

   The -i silent setting instructs the installer to run in the unattended installation mode.

> **Note:** If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

Example:

```
ca-sm-wss-<SVMVER>-cr-win32.exe -f "C:
\Program Files\CA\Web Services Security\install_config_info\ca-wss-installer.
properties" -i silent
```

An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-On Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

**Installation Notes:**

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_CA Single Sign-on_Web_Services_Security_Install_*install-date-and-time*.log
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
    **Default:** C:\Program Files\CA\Web Services Security
  - ***install-date-and-time***
    Specifies the date and time that the CA Single Sign-on WSS Agent was installed.
- The Agent cannotcommunicate properly with the Policy Server until the trusted host is registered.
- To stop the installation manually, type Ctrl+C.

## Copy cryptojFIPS.jar to the WebSphere JRE

> If the installer displays a warning message stating that the cryptojFIPS.jar file is not present in the WebSphere JRE, you must manually copy the file into that location before you register the WSS Agent.

Copy cryptojFIPS.jar from the following location in the WSS Agent installation:

- **Windows**:*WAS_HOME*\lib\ext\thirdparty
- **UNIX**:*WAS_HOME*/lib/ext/thirdparty

To the following location in the WebSphere installation:

- **Windows**:*WAS_HOME*\java\jre\lib\ext
- **UNIX**:*WAS_HOMsoaE*/java/jre/lib/ext

### Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time*_InstallLog.log file located in *WSS_Home*\install_config_info.

- ***date-time***
  Specifies the date and time of the CA Single Sign-On Web Services Security installation.

## Configure the WSS Agent for WebSphere and Register a Trusted Host on Windows

**Contents**



Configure the CA Single Sign-on WSS Agent and register the system that hosts it as a trusted host using the CA Single Sign-On Web Services Security Configuration Wizard.

### Gather Information Required for CA Single Sign-on WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

**Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Run the CA Single Sign-on WSS Agent Configuration Wizard on Windows

You can configure your CA Single Sign-on WSS Agent and register a trusted host immediately after installing the CA Single Sign-on WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a CA Single Sign-on WSS Agent on your system.

**Follow these steps:**

1. Open the following directory on your web server:
   *WSS_Home*\install_config_info

- ***WSS_Home***
  Specifies the path to where CA Single Sign-On Web Services Security is installed.
  **Default:** C:\Program Files\CA\Web Services Security

2. Right-click ca-pep-config.exe, and then select Run as administrator.

   **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

   The WSS Agent Configuration Wizard starts.

3. Use gathered system and component information to configure the CA Single Sign-on WSS Agent and register the host.

   **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*\config. You can modify this file.

- ***agent_home***
  Is the installed location of the CA Single Sign-on WSS Agent.

## Modify the SmHost.conf File (Windows)

WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *agent_home*\config directory.

2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

> **Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

- **hostconfigobject**

  Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

  If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

  **Example:** hostconfigobject="*host_configuration_object*"

- **policyserver**

  Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

  "*IP_address*, *port*,*port*,*port*"

  The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

  To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

  Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

  > **Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

  **Default:***IP_address*, 44441,44442,44443

  **Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

  **Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

  Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

  **Default:** 60

  **Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool (Windows)

When you install a CA Single Sign-on WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home* \bin directory when you install a CA Single Sign-on WSS Agent.

- ***agent_home***
  Is the installed location of the CA Single Sign-on WSS Agent.

**To re-register a trusted host using the registration tool**

1. Open a command prompt window.
2. Enter the smreghost command using the following required arguments:

```
smreghost -i policy_server_IP_address:[port]
-u administrator_username -p Administrator_password
-hn hostname_for_registration -hc host_configuration_ object
```

**Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

**-i *policy_server_IP_ address:port***

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

**Default:** (ports) 44441,44442,44443

**Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555

**Example**: (IPv4 default ports) -i 127.0.0.1

**Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

**Example:** (IPv6 default ports) -i [2001:DB8::/32]**-u *administrator_username***

Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**
  Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

  COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

  If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

  **Default:** COMPAT

  **Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

## Register Multiple Trusted Hosts on One System (Windows)

You typically register only one trusted host for each machine where web servers and Agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each CA Single Sign-On client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by CA Single Sign-On Agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

  **Note:** If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads: "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

  **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

# Install a WSS Agent for WebSphere on UNIX

This section contains the following topics:



## Prepare the Java Environment for the WSS Agent for WebSphere on UNIX

**Contents**



### Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

**To set the JRE in the PATH variable**

1. Open a Command Window.
2. Run the following commands:

```
PATH=$PATH:JRE
export PATH
```

   - ***JRE***
     Defines the location of your Java Runtime Environment bin directory.

### Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package.

The WebSphere JRE is based on Sun's JRE on the Solaris platform; this patch is available at Sun's website. The patch for other platforms is available at IBM's website. See the IBM documentation for more details.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows
  *WAS_HOME*\java\jre\lib\security

- UNIX
  *WAS_HOME*/java/jre/lib/security

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:

   - *JVM_HOME*\jre\lib\security (Windows)

   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.

2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.

3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

# Install the WSS Agent for WebSphere on UNIX

**Contents**

- Installation and Configuration Log Files (see page 857)

## Run the Installer to Install a WSS Agent Using a GUI

Install the WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-sm_version-unix_version.bin
  ```

  - ***sm_version***
    Specifies the version and, if applicable, the cumulative release number. The base version does not include a cumulative release number in the file name.

  - ***unix_version***
    Specifies the UNIX version: **sol** or **linux**.

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**Follow these steps:**

1. Exit all applications that are running.

2. Open a shell and navigate to where the install program is located.

3. Enter the following command:

   ```
   ./ca-sm-wss-sm_version-unix_version.bin
   ```

   The CA Single Sign-On Web Services Security installer starts.

4. Use gathered system and component information to install the WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agentsfor Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for IBM WebSphere**.

   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

   - When prompted for the location where WebSphere is installed, enter the correct location for your version of WebSphere.

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

   - Do not use space characters in the WSS Agent install path. For example, "/CA Technologies /agent" will result in install failure.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   **Note:** If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location. Afterward, the CA Single Sign-On Web Services Security Configuration screen is displayed.

6. Select one of the following options:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.

7. Click Done.
   If you selected the option to configure WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/*install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Run the Installer to Install a WSS Agent Using a UNIX Console

Install the CA Single Sign-on WSS Agent using the CA Single Sign-On Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-sm_version-unix_version.bin
  ```

- If you execute the CA Single Sign-On Web Services Security installer across different subnets, it can crash. Install CA Single Sign-On Web Services Security components directly on the host system to avoid the problem.

**Follow these steps:**

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.
3. Enter the following command:

   ```
   ./ca-sm-wss-sm_version-unix_version.bin -i console
   ```

   The CA Single Sign-On Web Services Security installer starts.
4. Use gathered system and component information to install the WSS Agent. Consider the following as you make your selections:

   - When prompted to select what agents to install, select **CA Single Sign-On Web Services Security Agentsfor Application Servers** and then specify the **CA Single Sign-On Web Services Security Agent for IBM WebSphere**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - When prompted for the location where WebSphere is installed, enter the correct location for your version of WebSphere.
   - Do not use space characters in the WSS Agent install path. For example, "/CA Technologies /agent" will result in install failure.
5. Review the information presented on the Pre-Installation Summary page, then proceed.

   Note: If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The WSS Agent files are copied to the specified location. Afterward, the CA Single Sign-On Web Services Security Configuration screen is displayed.
6. Select one of the following options:

   - Yes. I would like to configure CA Single Sign-On Web Services Security Agents now.
   - No. I will configure CA Single Sign-On Web Services Security Agents later.
7. Hit Enter.
   If you selected the option to configure WSS Agents now, the installation program prepares the CA Single Sign-On Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.
   If you did not select the option to configure WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/*install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

## Install a WSS Agent Using the Unattended Installer

After you have installed one or more WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home/*install_config_info

**To run the installer in the unattended installation mode**

1. From a system where CA Single Sign-On Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.
2. Copy the WSS Agent installer file (ca-sm-wss-*sm_version-unix_version*) into the same local directory as the ca-wss-installer.properties file.
   - ***sm_version***
     Specifies the CA Single Sign-Onrelease and, if applicable, cumulative release number. The base 12.52 SP1 release does not include a cumulative release number.
   - ***unix_version***
     Specifies the UNIX version: **sol** or **linux**.
3. Open a console window and navigate to the location where you copied the files.
4. Run the following command:

   ```
   ./ca-sm-wss-sm_version-unix_version -f ca-wss-installer.properties -i silent
   ```

   The -i silent setting instructs the installer to run in the unattended installation mode.

> **Note:** If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

Example:

```
./ca-sm-wss-sm_version-unix_version -f ~/CA/Web_Services_Security
/install_config_info/ca-wss-installer.properties" -i silent
```

An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-On Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

**Installation Notes:**

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/* install_config_info directory. This log file contains the results of the installation.
  - ***WSS_Home***
    Specifies the path to where CA Single Sign-On Web Services Security is installed.
  - ***install-date-and-time***
    Specifies the date and time that the WSS Agent was installed.
- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.
- To stop the installation manually, type Ctrl+C.

## Copy cryptojFIPS.jar to the WebSphere JRE

> If the installer displays a warning message stating that the cryptojFIPS.jar file is not present in the WebSphere JRE, you must manually copy the file into that location before you register the WSS Agent.

Copy cryptojFIPS.jar from the following location in the WSS Agent installation:

- **Windows**:*WAS_HOME*\lib\ext\thirdparty
- **UNIX**:*WAS_HOME*/lib/ext/thirdparty

To the following location in the WebSphere installation:

- **Windows**:*WAS_HOME*\java\jre\lib\ext
- **UNIX**:*WAS_HOMsoaE*/java/jre/lib/ext

## Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time* _InstallLog.log file located in *WSS_Home*\install_config_info.

- ***date-time***
  Specifies the date and time of the CA Single Sign-On Web Services Security installation.

# Configure the WSS Agent for WebSphere and Register a Trusted Host on UNIX

**Contents**



Configure a WSS Agent and register the system that hosts it as a trusted host using the CA Single Sign-On Web Services Security Configuration Wizard.

## Gather Information Required for WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

**Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Run the WSS Agent Configuration Program on UNIX or Linux Systems

You can configure your WSS Agents and register a trusted host immediately after installing the WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a WSS Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.

- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

**To configure Agents and register a trusted host**

1. If necessary, start the Configuration Wizard as follows:
   a. Open a console window.
   b. Navigate to *agent_home*/install_config_info, where *agent_home* is the installed location of the WSS Agent.
   c. Enter one of the following commands:
   GUI Mode: ./ca-pep-config.bin
   Console Mode: ./ca-pep-config.bin -i console

   The Configuration Wizard starts.
2. Use gathered system and component information to configure the WSS Agent and register the host.

   **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*/config. You can modify this file.

- ***agent_home***
  Is the installed location of the WSS Agent

## Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time*_InstallLog.log file located in *WSS_Home*\install_config_info.

- ***date-time***
  Specifies the date and time of the CA Single Sign-On Web Services Security installation.

## Modify the SmHost.conf File

WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *agent_home*/config directory.
   - ***agent_home***
     Is the installed location of the WSS Agent.
2. Open the SmHost.conf file in a text editor.
3. Enter new values for the any of the following settings that you want to change:

   **Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

   - **hostconfigobject**

     Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

     If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

     **Example:** hostconfigobject="*host_configuration_object*"

   - **policyserver**

     Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

     "*IP_address*, *port*,*port*,*port*"

     The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

     To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

     Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

**Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

**Default:***IP_address*, 44441,44442,44443

**Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

**Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

  Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

  **Default:** 60

  **Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool (UNIX)

When you install a CA Single Sign-on WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home* /bin directory when you install a WSS Agent.

- ***agent_home***
  Is the installed location of the WSS Agent.

**To re-register a trusted host using the registration tool**

1. Open a command prompt window.
2. Ensure that the library path environment variable contains the path to the agent bin directory.
3. Enter the following two commands:
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:*agent_home*/bin
   export LD_LIBRARY_PATH
   For example, enter the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/Web Services Security/wasagent/bin
   export LD_LIBRARY_PATH
   ```

4. Enter the smreghost command using the following required arguments:

   ```
   smreghost -i policy_server_IP_address:[port]
   -u administrator_username -p Administrator_password
   -hn hostname_for_registration -hc host_configuration_ object
   ```

   **Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

   **Note:** If the "-p Administrator_password" argument is not specified in the smreghost command, you are prompted to specify the password.

   ```
   smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
   -hc DefaultHostSettings
   ```

   The following example contains the -o argument:

   ```
   smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
   -hc DefaultHostSettings -o
   ```

   The following arguments are used with the smreghost command:

**-i *policy_server_IP_ address:port***

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

**Default:** (ports) 44441,44442,44443

**Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555

**Example**: (IPv4 default ports) -i 127.0.0.1

**Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

**Example:** (IPv6 default ports) -i [2001:DB8::/32]**-u *administrator_username***

Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

  Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**
  Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

- ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

**Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

**Default:** COMPAT

**Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

## Register Multiple Trusted Hosts on One System (UNIX)

You typically register only one trusted host for each machine where web servers and Agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each CA Single Sign-On client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by CA Single Sign-On Agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  **Note:** If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads:

  "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

# Uninstall a WSS Agent for WebSphere

To uninstall a CA Single Sign-on WSS Agent, run the CA Single Sign-On Web Services Security uninstall wizard.

**Follow these steps:**

1. Navigate to the *WSS_HOME*\install_config_info (Windows) or *WSS_HOME*/install_config_info (UNIX) directory and run the CA Single Sign-On Web Services Security uninstall wizard to remove CA Single Sign-On Web Services Security agents:

   - Windows: soa-uninstall.cmd
   - UNIX: soa-uninstall.sh
   - ***WSS_HOME***
     Specifies the CA Single Sign-On Web Services Security installation location.

   **Important!** If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA Single Sign-On Web Services Security Release Notes.

   The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.

3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.
   The uninstall wizard removes all selected CA Single Sign-On Web Services Security components.

4. Restart the server.

# Specify WSS Agent for IBM WebSphere Configuration Settings

**Contents**



## How to Configure the WSS Agent

To configure the WSS Agent, you must specify the following:

- Host Configuration Object (one for each host server)
- Agent Configuration Object (one for each WSS Agent)
- Agent identity (one for each WSS Agent)

**Follow these steps:**

1. On the Policy Server:

   a. Duplicate or create a Host Configuration Object, which holds initialization parameters for a Trusted Host.
   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

   b. As necessary, add or edit parameters in the Host Configuration Object that you just created.

   c. Duplicate or create an Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

   d. Add or edit required Agent parameters in the Agent Configuration Object.
   The configuration object must include the DefaultAgentName or AgentName parameter to specify the Agent identity.

   e. Create an Agent identity for the WSS Agent. You must select *Web Agent* as the Agent type for a WSS Agent.

2. On the system where the WSS Agent is installed:

    a. Run the Agent Configuration Wizard, which registers the Trusted Host.

    b. Enable the WSS Agent by setting the EnableWebAgent parameter in the Agent configuration file to Yes.

## WSS Agent for WebSphere Configuration File

By default, the WSS Agent for WebSphere installation creates a single agent configuration file, JavaAgent.conf. The agent configuration file is located in the *WSS_Home*/config directory.

- ***WSS_Home***
  Specifies the location where the WSS Agent is installed.

Each Agent configuration file is created with the following required default configuration parameters /values:

| Parameter | Description |
|---|---|
| DefaultAgentName | The agent identity the Policy Server uses to associate policies with the WSS Agent. The default value is "SoaAgent". Do not change this value. |
| EnableAgent | Specifies whether the WSS Agent is enabled. Possible values are Yes and No. Default value is Yes. |
| AgentConfigObject | The Agent Configuration Object specified during installation. |
| SmHostFile | Path to the local Host Configuration File. Path can be specified in absolute terms or relative to *WSS_HOME.*<br>**Note:** On Windows, specify paths using double backslashes ("\\") rather than single backslash ("\") to separate directories. On UNIX, use standard single slash ("/") separators.<br>Example values:<br>(Windows) C:\\Program Files\\CA\\Web Services Security\\wasagent\\config\\SmHost.conf<br>(Windows) config\\SmHost.conf<br>(UNIX) /config/SmHost.conf |
| ServerName | A string that will be used in the WSS Agent log to identify the WebSphere Server. |
| appserverjaasloginhandler | Specifies the Application Server-specific WSS Agent handler class for WebSphere. Default value is "com.ca.soa.agent.appserver.jaas.was.WasLoginHandler". Do not change this value. |

You need only edit the preconfigured values if the location of the Host Configuration File changes or you want to refer to a different Agent Configuration Object. If you use local configuration, you can add other Agent configuration parameters to these preconfigured values.

**Note:** Parameters that are held in the Agent configuration file are static. If you change these settings while the WebSphere server is running, the WSS Agent does not pick up the change until WebSphere is restarted.

The JavaAgent.conf file also contains a list of WSS Agent plugin classes; you do not need to alter this information.



**Note**: Leading and trailing whitespace in JavaAgent.conf value definitions is ignored. To include leading or trailing whitespace, quote the value (with either single or double quotes). Embedded, escaped quotes are unescaped during processing.

**Sample JavaAgent.conf (Windows)**

```
# SiteMinder WSS Agent Configuration File
#
# This file contains bootstrap information required by
# the SiteMinder WSS Agent
#
defaultagentname=SoaAgent
enableagent=yes
agentconfigobject=wsagent1_ac
servername=SOAWAS61
smhostfile=config\\SmHost.conf
appserverjaasloginhandler=com.ca.soa.agent.appserver.jaas.was.WasLoginHandler

# Configure plugins for the agent SoaAgent
transport_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.jaxrpcplugin.pluginconfig.JaxRpcPluginConfig
msg_body_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
credential_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
variable_resolver_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig

# <EOF>
```

## Agent Configuration Object

An Agent Configuration Object is a <stmdnr> policy object that holds Agent parameters for an Agent when using central agent configuration.



**Note:** Parameters held in an Agent Configuration Object are dynamic; if you change these settings while the WebSphere server is running, the WSS Agent will pick up the change.

## WSS Agent Configuration Parameters

The following table contains a complete list of all Agent configuration parameters supported by WSS Agents for Application Servers.

Unless otherwise noted,you can define parameters in either the Agent Configuration Object or the Agent configuration file depending upon how you decide to configure the WSS Agent.

| Parameter Name | Value | Description |
|---|---|---|
| AcceptTPCookie | YES or NO | (Optional) If set to yes, configures the WSS Agent to assert identities from third-party CA Single Sign-on session cookies (that is, session cookies generated by custom Agents created using the CA Single Sign-on and CA Single Sign-on WSS SDKs.<br>**Note:** AcceptTPCookie must be set to Yes to assert identities from session cookies generated by CA SOA Security Gateway.<br>Default is Yes. |
| AllowLocalConfig (Applies only in the Agent Configuration Object) | YES or NO | If set to yes, parameters set locally in the Agent configuration file take precedence over parameters in the Agent Configuration Object.<br>Default is NO. |
| AuthCacheSize | Number | (Optional) Size of the authentication cache for the WSS Agent (in number of entries). For example:<br>`authcachesize="1000"`<br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| AzCacheSize | Number | (Optional) Size of the authorization cache (in number of entries) for the WSS Agent. For example:<br>`authcachesize="1000"`<br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| CacheTimeout | Number | (Optional) Number of seconds before cache times out. For example:<br>`cachetimeout="1000"`<br>Default is 600 (10 minutes). |
| ConfigObject (Applies only in Agent configuration file) | String | The name of the Agent Configuration Object associated with the WSS Agent.<br>No default value. |
| CookieDomain | String | (Optional) Name of the cookie domain. For example:<br>`cookiedomain="ca.com"`<br>No default value.<br>For more information, see the cookiedomainscope parameter. |
| CookieDomainScope | Number | (Optional) Further defines the cookie domain for assertion of CA Single Sign-on session cookies by the WSS Agent. The scope determines the number of sections, separated by periods, that make up the domain name. A domain always begins with a period (.) character. For example:<br>`cookiedomainscope="2"` |

| Parameter Name | Value | Description |
|---|---|---|
| | | Default is 0, which takes the domain name specified in the cookiedomain parameter. |
| DefaultAgentName (Applies only in the Agent Configuration Object) | String | The agent identity the Policy Server will use to associate policies with the WSS Agent.<br>Default is "SoaAgent"; this value should not changed. |
| EnableWebAgent (Applies only in Agent configuration file) | YES or NO | Enables or disables the WSS Agent. When set to 'yes', the WSS Agent will protect resources using the Policies configured in the Policy Server for the configured agent identity.<br>Default is Yes. |
| LogOffUri | String | (Optional) The URI of a custom HTTP file that will perform a full log off (removing the session cookie from a user's browser). A fully qualified URI is not required. For example, LogOffUri could be set to: /Web pages/logoff.html<br>No default value. |
| PsPollInterval | Number | (Optional) The frequency with which the WSS Agent polls the Policy Server to retrieve information about policy changes.<br>Default is 30 seconds. |
| ResourceCacheSize | Number | (Optional) Size (in number of entries) of the cache for resource protection decisions. For example:<br>`resourcecachesize="1000"`<br><br>Default is 2000.<br>To flush this cache, use the Administrative UI. |
| SAMLSessionTicketLogoffi | YES or NO | (Optional) Determines whether the WSS Agent should attempt to log off session tickets in SAML assertions.<br>Default is Yes. |
| ServerName (Applies only in Agent configuration file.) | String | A string to be used in the WSS Agent log to identify the target application server. |
| SessionGracePeriod | Number | (Optional) Grace period (in seconds) between the regeneration of session tokens.<br>Default is 30 |
| SmHostFile (Applies only in Agent configuration file) | String | Path to the local Host Configuration File (typically *WSS_Home*\conf\SmHost.conf).<br>No default value. |
| XMLAgentSoapFaultDetails | YES or NO | |

| Parameter Name | Value | Description |
|---|---|---|
| | | (Optional) Determines whether or not the WSS Agent should insert the authentication/authorization rejection reason (if provided by the Policy Server) into the SOAP fault response sent to the web service consumer.<br>Default is No. |
| XMLSDKAcceptSMSessionCookie | YES or NO | (Optional) Determines whether or not the WSS Agent accepts an CA CA Single Sign-on session cookie to authenticate a client.<br>Default is No.<br>If set to Yes, the WSS Agent uses information in a session cookie sent as an HTTP header in the request as a means of authenticating the client.<br>If set to No, session cookies are ignored and the WSS Agent requests credentials required by the configured authentication scheme. |
| XMLSDKMimeTypes | String | (Optional) A comma-delimited list of MIME types that the WSS Agent will accept for processing by CA Single Sign-On Web Services Security. All POSTed requests having one of the listed MIME types are processed. Examples:<br>text/xml<br>application/octet-stream<br>text/xml,multipart/related<br>If you do not add this parameter to the Agent Configuration Object, the WSS Agent defaults to accepting text/xml and application/soap+xml MIME types. |

## Configure the Username and Password Digest Token Age Restriction

By default, the WS-Security authentication scheme imposes a 60-minute restriction on the age of Username and Password Digest Tokens to protect against replay attacks.

To configure a different value for the token age restriction for a WSS Agent for Application Servers, set the WS_UT_CREATION_EXPIRATION_MINUTES parameter in the XmlToolkit.properties file for that agent.

**Follow these steps:**

1. Navigate to one of the following locations:
   - *WAS_HOME*\properties (Windows)
   - *WAS_HOME/*properties (UNIX)
     - ***WAS_HOME***
       Specifies the WebSphere install directory.

     For example, on Windows:

     ```
     C:\Program Files\WebSphere\AppServer\properties
     ```
2. Open XmlToolkit.properties in a text editor.
3. Uncomment and modify the WS_UT_CREATION_EXPIRATION_MINUTES parameter line to configure a different value for the token age restriction:

   ```
   WS_UT_CREATION_EXPIRATION_MINUTES=token_age_limit
   ```

- ***token_age_limit***
  Specifies the token age limit restriction in minutes.

4. Save and close the XmlToolkit.properties file.

5. Restart the WSS Agent.

# Configure WebSphere to Work with the WSS Agent

**Contents**



## Set the JAVA_AGENT_ROOT JVM System Property

Because the WSS Agent may not be installed in the same file system location on every system in clustered and SSO WebSphere environments, you must define a JVM system property, JAVA_AGENT_ROOT to define the installed location of the WSS Agent.

**To set the JAVA_AGENT_ROOT JVM system property**

1. Open the WebSphere Integrated Solutions Console.

2. Click the following, in the order shown:
   In the navigation tree: Servers, Application Server
   In the work area: *server_name*, Java and Process Management, Process Definition, Java virtual Machine, Additional Properties, Custom Properties.

3. Create a new variable in Custom Properties named JAVA_AGENT_ROOT and specify its value as the location where the WSS Agent is installed. For example, in Windows enter:

   ```
   JAVA_AGENT_ROOT=C:\SoaSecurityManager\wasagent
   ```

4. Save the changes in the master repository.

## Set the log.log-config-properties Environment Variable

You must define a JVM system property, log.log-config-properties, to define the location of the WSS Agent logging configuration file.

**To set the log.log-config-properties JVM system property**

1. Open the WebSphere Integrated Solutions Console.

2. Click the following, in the order shown:
   In the navigation tree: Servers, Application Server
   In the work area: *server_name*, Java and Process Management, Process Definition, Java Virtual Machine, Additional Properties, Custom Properties.

3. Create a new variable in Custom Properties named log.log-config-properties and specify its value as the location of the WSS Agent logging configuration file (relative to the installed location of the WSS Agent, *WSS_HOME)*.
   For example, in Windows enter:

   ```
   log.log-config-properties=config\log-config.properties
   ```

4. Save the changes in the master repository and restart the server.

## Configure General WebSphere Settings

Before you configure the WSS Agent, you must do the following:

- Configure the active user registry for security
- Enable WebSphere Global Security
- Enable Security Attribute Propagation for WebSphere SSO, if required

### Enable WebSphere Security Options

**To enable security options for the WebSphere managed domain**

1. If necessary, start the WebSphere Server and the WebSphere Integrated Solutions Console.
2. In the navigation tree click one of the following as appropriate for your WebSphere version:
   - WebSphere 6.x: Security, Secure administration, applications, and infrastructure
   - WebSphere 7.x: Security, Global Security, Java 2 Security
3. Set the Enable Administrative Security option.
4. Set the Use Java 2 security to restrict application access to local resources option.
5. Click Apply to apply your changes. To save changes, click System Administration and Save Changes to Master Repository.
   a. **Note:** Until you save changes to the master repository, the Integrated Solutions Console uses a local workspace to track your changes.

### Configure LDAP as a WebSphere User Registry

In a typical deployment, WebSphere and the Policy Server are configured to use the same LDAP user registry.

**Note:** If you are not configuring WebSphere and the Policy Server to use the same LDAP user registry (typically because WebSphere is already configured with a custom user registry), verify that the custom registry is properly configured (see the WebSphere documentation for information) and configure user mapping.

**To configure a Policy Server LDAP user directory as a WebSphere user registry**

1. If necessary, start the WebSphere Server and the WebSphere Integrated Solutions Console.
2. In the navigation tree click one of the following as appropriate for your WebSphere version:
   - WebSphere 6.x: Security, Secure administration, applications, and infrastructure
   - WebSphere 7.x: Security, Global Security, User Account Repository
3. In the User account repository section, select Standalone LDAP Registry from the Available Realm Definitions drop-down menu.
4. Click Apply to save your changes.
5. Click Configure.
6. Under Server user identity, enter the select the Server identity that is stored in repository option and type the identity and password of a user account to use to run the application server for security purposes in the corresponding fields.
7. Under General Properties , fill in the following fields and then click Apply.
   - Server user ID
   - Server user Password
   - Type
   - Host
   - Port
   - Base Distinguished Name (DN)
   - Bind Distinguished Name (DN)
   - Bind Password
   - Search timeout
8. Depending on the WebSphere configuration, check Reuse Connection and Ignore case for authorization.
9. On WebSphere 7.0, select the Standalone LDAP registry option from the Available realm definitions drop-down and click Set as current.

10. Click Apply to apply your changes. To save changes to the master repository, click System Administration and Save Changes to Master Repository.

    a. **Note:** Until you save changes to the master repository, the Integrated Solutions Console uses a local workspace to track your changes.

## Configure the WSS Agent Login Module in WebSphere

You configure the WSS Agent Login Module in the WebSphere Application Server using the WebSphere Integrated Solutions Console. General information about configuring Login Modules is available in the WebSphere documentation.

**To configure the WebSphere Application Server to use the WSS Agent Login Module**

1. If necessary, start the WebSphere Server and the WebSphere Integrated Solutions Console.

2. Click the following, in the order shown:
   In the navigation tree: Security, Secure Administration, Applications and Infrastructure.
   In the work area: Java Authentication and Authorization Service, System Logins.

3. Click New to create a new System Login profile. This profile will contain WSS Agent Login Module and two other standard WebSphere login modules create the WebSphere identity and credentials so that the identity is propagated to the rest of WebSphere and can be used for WebSphere single sign-on.

4. Under General Properties on the New page, enter "XMLAgent" in the Alias field and click Apply.

5. Under Additional Properties, click JAAS login modules.

6. Add the WSS Agent Login Module:

   a. On the JAAS Login Modules page, click New.

   b. Under General Properties on the New page, enter the WSS Agent Login Module class name:

      ```
      com.ca.soa.agent.appserver.jaas.XMLAgentLoginModule
      ```

   c. Ensure that REQUIRED is selected from the Authentication strategy drop-down list.

   d. Click Apply to save your changes.

7. Add the WebSphere LTPA Login Module:

   a. Back on the JAAS Login Modules page, click New.

   b. Under General Properties on the New page, enter the WebSphere LTPA Login Module class name:

      ```
      com.ibm.ws.security.server.lm.ltpaLoginModule
      ```

   c. Ensure that REQUIRED is selected from the Authentication strategy drop-down list.

d. Click Apply to save your changes.

8. Add the WebSphere Default Inbound Login Module:

   a. Back on the JAAS Login Modules page, click New.

   b. Under General Properties on the New page, enter the WebSphere Default Inbound Login Module class name:

   c. com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule

   d. Ensure that REQUIRED is selected from the Authentication strategy drop-down list.

   e. Click Apply to save your changes.

9. Back on the JAAS Login Modules page, click Set Order.

10. Under General Properties on the JAAS Login Module Order page, if necessary, move the Login Modules so that they appear in the following order:

```
com.ca.soa.agent.appserver.jaas.XMLAgentLoginModule
com.ibm.ws.security.server.lm.ltpaLoginModule
com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule
```

11. Click Apply to save your changes.To save changes permanently, click System Administration and Save Changes to the Master Repository.

**Note:** Until you save changes to the master repository, the Integrated Solutions Console uses a local workspace to track your changes.

# WSS Agent for IBM WebSphere Logging

**Contents**



## log4j

The WSS Agent logger for application servers is implemented using Apache's log4j. For more information, see http://logging.apache.org/log4j/docs/.

# Log Files

Two log files provide important information about the WSS Agent:

- WSS Agent log file—Logs WSS Agent error and processing messages.
- WSS Agent XML message processing log file—Logs messages information relating specifically to the WSS Agent's processing of XML messages

## WSS Agent Log

This WSS Agent writes information about its standard operations and performance to the WSS Agent log.

By default, WSS Agent logging is enabled and written to the XmlAgent.log file in:

- Windows—*WSS_Home*\wasagent\log
- UNIX—*WSS_Home*/wasagent/log

You can change WSS Agent logging parameters by editing the log-config.properties file located in:

- Windows—*WSS_Home*\wasagentconfig\
- UNIX— *WSS_Home*/wasagent/config/

**Note:** These are the default values; the logging configuration file name and location can be changed by editing the log.log-config-properties JVM system property.

## WSS Agent XML Message Processing Logging

In addition to its standard logging functionality, WSS Agents for IBM WebSphere also log information relating specifically to their processing of XML messages. Like the WSS Agent log, the XML message processing log is also implemented using Apache's *log4j* standard.

**Note:** WSS Agent XML message processing logging does not start until an XML message that needs to be processed is received.

By default, WSS Agent XML message processing logging is enabled and written to the soasm_agent. log file in:

- Windows—*WSS_Home*\wasagent\bin\
- UNIX—*WSS_Home*/wasagent/bin/

You can change WSS Agent XML message processing logging parameters by editing the log.config file, which can be found in:

- Windows—*WSS_Home*\wasagent\config\
- UNIX— *WSS_Home*/wasagent/config/

## Change the WSS Agent Log File Name

To change pathname of the WSS Agent log file, edit the log.logfile-pattern parameter. Possible values are valid pathnames. If you specify a relative value, the path is set relative to the JAVA_AGENT_ROOT JVM system property.

Default value: "log\XmlAgent.log"

For example:

```
log.logfile-pattern=log\XmlAgent.log
```

## Append Messages to an Existing WSS Agent Log File

To add logging information to an existing WSS Agent log file instead of rewriting the entire file each time logging is invoked, add the log.logfile-append-on-reset parameter.

For example:

```
log.logfile-append-on-reset=YES
```

## Set the WSS Agent File Log Level

To change the WSS Agent log level, edit the log.logging-level parameter. Possible values are:

- DEBUG - Logs all; most verbose
- CONFIG - Configuration information
- INFO - Information
- WARNING - Warnings
- SEVERE - Errors only; least verbose

Default value: WARNING

For example:

```
log.logging-level=INFO
```

## Roll Over the WSS Agent Log File

To change file size limit at which the WSS Agent log should rollover, change the log.logfile-limit parameter. Rolling over a log file starts a new log file, preventing a single log file from becoming unmanageable. Possible values are numbers, representing kilobytes.

The default value is 1000.

For example:

```
log.logfile-limit=512
```

## Disable WSS Agent XML Message Processing Logging

To disable WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent config subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

## WSS Agent Log Configuration File Summary

The WSS Agent logging configuration file defines default WSS Agent logging settings.

Available configuration parameters are:

| Name | Description |
|---|---|
| log.logfile-append-on-reset | Add logging information to an existing log file instead of creating a new file each time logging is invoked.<br>Default value: no |
| log.logfile-pattern | Specifies the pathname (relative to *WSS_Home*) of the WSS Agent log file.<br>Default value: log/XmlAgent.log |
| log.logging-level | Defines the logging level. The levels are:<br>DEBUG - all logging, most verbose<br>CONFIG - configuration information<br>INFO - information<br>WARNING -warnings<br>SEVERE - errors<br>Default value: WARNING |
| log.logfile-limit | Specifies the size limit, in kilobytes Rollover a log file after it reaches the specified size.<br>Default value: 1,000KB |

**Note:** Once the WSS Agent connects to the Policy Server, corresponding logging settings found in the Agent Configuration Object override the values in log-config.properties.

# Finalize the WSS Agent for WebSphere Installation

**Contents**

## Restart WebSphere

After completing WebSphere-side configuration of the WSS Agent, you must restart WebSphere.

**To restart IBM WebSphere**

1. Log out of the Integrated Solutions Console.

2. From a command line or shell in the *WAS_HOME*/bin directory, stop and then restart the WebSphere Server.
   To stop the server, you will require the server user ID and Server user password you entered when configuring LDAP as a WebSphere user registry. The command is:

   ```
   stopServer server1 -username serveruserID -password serveruserpassword
   ```

   To start the server, you do not need a password:

   ```
   startServer server1
   ```

3. To make sure everything is working as expected, view the WSS Agent and WebSphere (SystemOut.log, SystemErr.log) log files.
   WebSphere's SystemOut.log and SystemErr.log file resides in:

   ```
   WAS_HOME/profiles/profile_name/logs/server_name
   ```

   The logs indicate should indicate that everything is working correctly. If the logs indicate problems, you should troubleshoot your configuration.

## Edit Deployment Descriptors of JAX-RPC Applications

To protect a JAX-RPC web service you must edit its deployment descriptor to add the WSS Agent JAX-RPC Handler.

**To edit a JAX-RPC web service deployment descriptor**

1. Unpack the enterprise archive (EAR) containing one or more web services.

2. Examine the EAR to determine which of the modules within it contains a JAX-RPC web service. (A module that contains a JAX-RPC web service if it has a webservices.xml file in the META-INF folder for EJB endpoints, or the WEB-INF folder for servlet endpoints.)

3. For each module in the EAR identified as a JAX-RPC web service:

   a. Unpack the archive containing the module. (The archive will be a JAR file for EJB endpoints and a WAR file for servlet endpoints.)

   b. Find the webservices.xml file.

   c. For each port-component element found in the webservices.xml file, add a handler element:

   ```
   <handler>
      <handler-name>SiteMinder WSS Agent Handler</handler-name>
      <handler-class>
         com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler
   ```

```
   </handler-class>
</handler>
```

**Note:** The WSS Agent JAX-RPC handler must always be invoked first; If other handler elements are already present or subsequently added to the webservices.xml file, the WSS Agent JAX-RPC Handler element must placed before them.

4. Repackage the module into the appropriate archive type (JAR or WAR).
5. When all modules have been configured, repackage the EAR.
6. Install or update the enterprise application.

### Configure Policies for the WSS Agent

You create authentication and authorization policies to protect web service resources hosted on WebSphere from their associated WSDL files using the CA Single Sign-On Web Services Security Configuration User Interface.

# Web Agent Option Pack

The following sections detail how to install the Web Agent Option Pack and deploy the Federation web services.

## Web Agent Option Pack Features

The Web Agent Option Pack is a stand-alone component that must be installed separately from CA Single Sign-On. The component installs Federation Web Services (FWS) and support for eTelligent Rules POST variables on an application server or web server.

**Note:** Unlike the Web Agent Option Pack, the Policy Server Option Pack is no longer a stand-alone component. The Policy Server Option pack is included with the Policy Server installation.

The Web Agent Option Pack supports the following features:

- CA Single Sign-On Federation (Partnership and Legacy)
  - **Partnership Federation**
    Partnership federation is based on configuring federated partnerships. The partnership model does not require configuration of CA Single Sign-On–specific objects, such as domains, realms, and policies.

- **Legacy Federation**
  Legacy Federation (formerly Federation Security Services).
  Legacy federation is based on configuring CA Single Sign-On objects, such as affiliate domains, authentication schemes, and policies to protect federated resources.
  For more information, see CA Single Sign-On Federation: Legacy Federation.

- **eTelligent Rules**
  eTelligent Rules are policy expressions that combine Boolean operators and user-defined variables and that are evaluated at runtime. As policy expressions, eTelligent Rules allow administrators to implement fine-grained access control of protected resources on a Policy Server–protected website. To support POST variables, the Web Agent Option Pack is required.

# Web Agent Option Pack Installation Requirements

**Contents**



## General Option Pack Installation Requirements

Before you install the Web Agent Option Pack, the following components are required:

- Supported application server
  For the supported application servers, see the Platform Support Matrix at the Technical Support ( http://www.ca.com/support) site.
  If you use ServletExec as your application server, apply the most current hot fixes. Federation Web Services requires these hot fixes to work with ServletExec. To obtain the hot fixes, go to the New Atlanta Communications (http://www.newatlanta.com) web site.

- A supported Java Development Kit (JDK).
  This JDK is required even if you are using an application server that ships with a JDK or JRE.

- For Linux operating platforms, be sure that the required Linux libraries are installed.

You can install the Web Agent Option Pack without the Web Agent. However, install the Web Agent *before* using federation.

## Required Linux Libraries

Certain library files are required for components operating on Linux systems. Failure to install the correct libraries can cause the following error:

```
java.lang.UnsatisfiedLinkError
```

If you are installing, configuring, or upgrading on a Linux system, the following RPM packages are required on the host system:

**Red Hat 5.x:**

| | |
|---|---|
| compat - gcc-34-c++-3.4.6-patch_version.i386.rpm | libidn.so (http://libidn.so).11 |
| libstdc++-4.x.x-x.el5.i686.rpm | ncurses-5.5-24.20060715.rpm |

**Red Hat 6.x:**

| | |
|---|---|
| libstdc++-4.x.x-x.el6.i686.rpm | libidn.so (http://libidn.so).11 |
| libidn-1.18-2.el6.i686.rpm | ncurses-5.7-3.20090208.el6.i686.rpm |
| libXext.i686.rpm | ncurses-libs-5.7-3.20090208.el6.i686.rpm |
| libXrender.i686.rpm | ncurses-base-5.7-3.20090208.el6.i686.rpm |
| libXtst.i686.rpm | |

**Additional Packages for Red Hat 6.x (64-bit):**

In addition to the packages for Red Hat 6.x, these packages are also needed for 64-bit Red Hat.

**Note:** The following packages are required only if you install a 32-bit Web Agent on a 64-bit machine.

| | |
|---|---|
| libXau-1.0.5-1.el6.i686.rpm | libXext-1.1-3.el6.i686.rpm |
| libxcb-1.5-1.el6.i686.rpm | compat-libstdc++-33-3.2.3-69.el6.i686.rpm |
| compat-db42-4.2.52-15.el6.i686.rpm | compat-db-4.6.21-15.el6.i686.rpm |
| compat-db43-4.3.29-15.el6.i686.rpm | libXi-1.3-3.el6.i686.rpm |
| libX11-1.3-2.el6.i686.rpm | libXtst-1.0.99.2-3.el6.i686.rpm |
| libXrender-0.9.5-1.el6.i686.rpm | libXft-2.1.13.4.1.el6.i686.rpm |
| libexpat.so (http://libexpat.so).1 (provided by expat-2.0.1-11.el6_2.i686.rpm) | libXt-1.0.7-1.el6.i686.rpm |
| libfreetype.so (http://libfreetype.so).6 (provided by freetype-2.3.11-6.el6_2.9.i686.rpm) | libXp-1.0.0-15.1.el6.i686.rpm |

| | |
|---|---|
| libfontconfig.so (http://libfontconfig.so).1 (provided by fontconfig-2.8.0-3.el6.i686.rpm) | libstdc++.i686.rpm |
| libICE-1.0.6-1.el6.i686.rpm | compat-libtermcap.i686.rpm |
| libuuid-2.17.2-12.7.el6.i686.rpm | libidn.i686.rpm |
| libSM-1.1.0-7.1.el6.i686.rpm | |

**Red Hat 7.x:**

The following packages are required to install 64-bit Web Agent Option Pack on Red Hat 7.x 64-bit:

- libstdc++-4.8.3-9.el7.i686
- libidn-1.28-3.el7.i686
- libXext-1.3.2-2.1.el7.i686
- libXrender-0.9.8-2.1.el7.i686
- libidn.so.11.rpm
- libXtst-1.2.2-2.1.el7.i686
- ncurses-5.9-13.20130511.el7.i686
- ncurses-libs-5.9-13.20130511.el7.i686
- ncurses-base-5.9-13.20130511.el7.i686

### Requirements for Windows System with JBoss (179105)

The Windows system with JBoss 5.1.2 where you install the Web Agent Option Pack must meet the following recommended system requirements:

- Memory—4 GB of system RAM.
- JVM heap size—Set the values for max heap and min heap as follows:
  - Max Heap - (-Xmx 1024m or -Xmx1g)
  - Min Heap - (-Xms 1024m or -Xms1g)

## System Locale Must Match the Language of Installation and Configuration Directories

To install and configure a CA Single Sign-On component to a non-English directory, set the system to the same locale as the directory. Also, make sure that you installed the required language packages so the system can display and users can type localized characters in the installer screens.

For the details on how to set locale and required language packages, refer to respective operating system documents.

## Components Required for CA Single Sign-On Federation

The following components are required for CA Single Sign-On Federation (Legacy and Partnership):

- Policy Server
- Application server or web server
  **Note:** An application server with built–in web server, such as JBOSS, WebLogic or WebSphere, is required to deploy Federation Web Services. Alternately, a web server with an application server plug-in, such as ServletExec, can be used.
- Web Agent Option Pack

**Note:** Refer to the *Federation Release Notes* for any known issues regarding federation features and the Web Agent Option Pack.

## Components Required for eTelligent Rules

The following components are required for eTelligent Rules:

- Policy Server
- Web Agent Option Pack
  The Web Agent Option Pack is required to support eTelligent Rules that contain POST variables.

## Version Compatibility

If the Web Agent and Web Agent Option Pack are installed on the same server, they must both be the *same* version, including the Service Pack and CR version. However, the Web Agent Option Pack can operate with Policy Servers of different, but compatible versions. To learn which Policy Server versions the Web Agent Option Pack is compatible with, see the CA Single Sign-on Platform Support Matrix.**Important!** When different compatible versions of the Web Agent Option Pack and Policy Server are mixed, federation functionality is limited to that of the lesser of the two components. Thus, if the Policy Server supports a federation feature but the Web Agent Option Pack does not, a request that requires that functionality may be rejected or the functionality ignored. The same is also true if the Web Agent Option pack supports a feature and the Policy Server does not.

## Environment Variables Added by the Installation

The installation of the Web Agent Option Pack sets the following environment variables:

- NETE_WA_OPACK = “INSTALLED”
- NETE_WA_PATH = $NETE_WA_ROOT$$/$bin;$NETE_WA_ROOT$$/$bin$ /$thirdparty;$NETE_JRE_ROOT$$/$bin;$NETE_JRE_ROOT$$/$bin$/$server

## Java Virtual Machine Installation Error on Solaris can be Ignored (149886)

**Symptom:**

You are doing a console mode installation of a CA Single Sign-On product on a Solaris platform. The following error message displays: "Unable to install the Java Virtual Machine included with this installer."

**Solution:**

Ignore this error message. The error is a third-party issue and it has no functional impact.

## Web Agent Option Pack on JBOSS Requires Workaround

**Symptom:**

On a JBoss 5.1.2 server, system JARs are overriding application-specific JARs, such as those JARs for the Web Agent Option Pack.

**Solution:**

Prevent the Web Agent Option Pack XML API files from being overwritten by JBOSS system JARS.

**Important!** This workaround only applies to the version of JBOSS 5.1.x that the product supports.

Add the following filter package in two places in the **war-deployers-jboss-beans.xml** file:

```
<property name="filteredPackages">javax.servlet,org.apache.commons.
logging,javax.xml.parsers,org.xml.sax,org.w3c.dom</property>
```

The filter package allows the use of the Web Agent Option Pack XML API files instead of the JBOSS system files.

**Follow these steps:**

1. Locate the war-deployers-jboss-beans.xml file located in the directory:
   /deployers/jbossweb.deployer/META-INF/

2. Find the following entry:

   ```
   <property name="filteredPackages">javax.servlet,org.apache.
   commons.logging</property>
   ```

3. Change the entry to:

   ```
   <property name="filteredPackages">javax.servlet,org.apache.commons.
   logging,javax.xml.parsers,org.xml.sax,org.w3c.dom</property>
   ```

   This entry in the file is on one line.

4. Find the second instance of the entry in step 2 and replace it with the entry in step 3. Add the filter package in both places in the XML file.

5. Save the XML file.

# Install the Web Agent Option Pack

**Contents**



## Installation Modes

The Web Agent Option Pack can be installed on a Web server running ServletExec or other supported application server, such as WebLogic, WebSphere, or JBoss.

You can install the Web Agent Option Pack in one of the following modes:

- GUI Mode
  GUI mode uses a graphical installation wizard to install the Web Agent Option Pack.

- Console Mode (UNIX platforms only)
  Uses command line questions about the installation in a UNIX console window.

- Unattended Mode
  Installs the Web Agent Option Pack without user intervention. Use the unattended installation mode to automate more installations on other machines in your network.

**Note:** You must install the Web Agent Option Pack using GUI or Console mode *before* running an unattended installation. The initial installation creates a properties file that contains the installation settings for the unattended installation.

## Run the Web Agent Option Pack Installer

The Web Agent Option Pack can be installed as a stand-alone product. The installer attempts to find an installed Web Agent, but if it cannot, it prompts you to continue or cancel. Continuing prompts you for an installation path. Then the installer installs the Option Pack in the location that you specify.

Install the Web Agent Option Pack using the method for your platform:

- Windows systems: install in GUI mode.

**Important!** If you are installing the Web Agent Option Pack on a Windows system immediately after installing the Web Agent, reboot your system first.

- UNIX systems: install in GUI or console mode.
  To install in console mode, you execute the Option Pack binary with the -i console command argument.

**Follow these steps:**

1. Stop the Web or application server and exit any applications that are running.
2. Log in to the CA Technical Support (http://www.ca.com/support) site.
3. Click Download Center.
4. Search the Download Center for the installation kit you need.
5. Run the installation program according to your platform.
   - **Windows:** Double-click the executable.
   - **UNIX:** At the command prompt, type one of the following commands:
     - **GUI Mode**
       *./binary_filename*
     - **Console Mode**
       *./binary_filename* -i console

     **Example:** To run the installation in GUI mode on a Solaris platform, enter: **./ca-wa-opack-12.52-sol.bin**

     **Note:** If needed, use the chmod command to add execute permissions to the installation file, for example:

     ```
     chmod +x ca-wa-opack-12.52-sol.bin
     ```
6. Follow the installation dialogs and prompts to complete the installation.

To re–install the Option Pack, run the executable again.

## Move smvariable.dll File for eTelligent Rules on Windows

After you install the Web Agent Option Pack on your web server, move the smvariable.dll files if your environment meets *all* the following criteria:

- You want to use eTelligent rules.
- You are using the Windows 2008 R2 (64-bit) operating environment.

- You are using an IIS 7.x Web Agent.

Perform this procedure for each installation of your Web Agent Option Pack.

**Follow these steps:**

1. Move the dll file for 32-bit applications by performing the following steps:
    a. Locate the following file:

    ```
    C:\Program Files\CA\webagent\win32\bin\non_stub\smvariable.dll
    ```

    b. Move the previous file into the following directory:

    ```
    C:\Program Files\CA\webagent\win32\bin
    ```

2. Move the dll for 64-bit applications by performing the following steps:
    a. Locate the following file:

    ```
    C:\Program Files\CA\webagent\win64\bin\non_stub\smvariable.dll
    ```

    b. Move the previous file into the following directory:

    ```
    C:\Program Files\CA\webagent\win64\bin
    ```

The smvariable.dll files have been moved to accommodate eTelligent rules.

## Move smvariable.dll File for eTelligent Rules on Linux or UNIX

Move the smvariable.so library if you want to use the following criteria:

- eTelligent rules
- Post variables

Perform this procedure for each installation of your Web Agent Option Pack.

**Follow these steps:**

1. Move the smvariable.so by performing the following steps:
    a. Locate the following file:

    ```
    /webagent_optionpack/bin/libsmvariable.so
    ```

    b. Move the previous files into the following directory:

    ```
    /webagent/bin
    ```

The smvariable.so files have been moved to accommodate eTelligent rules and Post variables.

## Next Step After Installation

After you complete the Web Agent Option Pack installation, you can configure the features that you want.

**eTelligent Rules**

Configure eTelligent rules.

**Federation**

For CA Single Sign-On Federation, configure the components to establish successful federated partnerships. Most of these components are configurable using the Administrative UI.

# Deploy Federation Web Services

Contents



Federation Web Services (FWS) is a collection of servlets that are packaged as a web application in accordance with the Java Servlet API 2.3 specification. The Federation Web Services application is installed with the Web Agent Option Pack. The application is deployed within an application server, or deployed inside the Tomcat web server, which is embedded in the CA Access Gateway.

The web application is rooted at a specific URL on the web or application server, http://www.*your_server.com*/affwebservices/. URLs for the servlets included with the FWS application have this same root. The Federation Web Services application provides these services:

- Assertion Retrieval Service (SAML 1.x)
- SAML credential collector (SAML 1.x)
- Intersite Transfer Service (SAML 1.x)
- Artifact Resolution Service (SAML 2.0)
- Assertion Consumer Service (SAML 2.0)
- Security Token Consumer Service (WS-Federation)
- AuthnRequest service (SAML 2.0)
- Single Sign-on service (SAML 2.0 and WS-Federation)
- Single Logout Service (SAML 2.0)
- Signout Service (WS-Federation)

## Properties File for Federation Web Services

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. For deploying FWS, set only the parameter that specifies the location of the WebAgent.conf file. For the other settings, accept the default values or modify the values as needed.

Note: The AffWebServices.properties file is in UTF-8 format. If you plan to modify this file, use an editor that supports this format.

The settings are as follows:

| AffWebServices. properties Settings | Value |
|---|---|
| NotificationLibraryType | Specifies the library type the Web Agent uses for notification alerts.<br>Note: The CA Access Gateway does not support this setting. |
| NotificationLibraryDetails | Indicates the Java classname or the C library and function name.<br>Note: The CA Access Gateway does not support this setting. |
| SMserverPort | Determines which Policy Server service at the producer processes the notification tunnel calls. |
| AgentConfigLocation | Indicates the location of the WebAgent.conf file. You must specify the location of the configuration file. |

The installed location of the AffWebServices.properties file is in the following locations:

- For a web or application server
  *web_agent_or_webagent_option_pack_home*/affwebservices/WEB-INF/classes
- For the CA Access Gateway:
  *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes

*web_agent_home*
Indicates the installed location of the Web Agent.

*sps_home*
Indicates the installed location of CA Access Gateway.

## Agent Configuration Object Settings Used by FWS

For partnership federation, Federation Web Services (FWS), installed by the Web Agent Option Pack uses the following agent configuration object settings for federated communication. You configure agent configuration objects in the Administrative UI.

- defaultagentname

  Note: The FWS application uses the value of the defaultagentname parameter and not the agentname parameter.

- TransientIDCookies
- AcceptTPCookie

- TransientIPCheck
- CookieDomain
- CookieDomainScope
- SSOZoneName
- SSOTrustedZone
- UseSecureCookies

## Enable FWS Logging

The LoggerConfig.properties file lets you enable logging so the Federation Web Services application can record the following information:

- Assertion retrieval
- Session management
- Notification alert information
- Trace messages

The log file shows activity at the asserting party and the relying party, depending on how your site is configured.

Note: The LoggerConfig.properties file is in UTF-8 format. If you plan to modify this file, use an editor that supports this format.

The installed location of the LoggerConfig.properties file is:

- For the Web Agent, the location is *web_agent_home*/affwebservices/WEB-INF/classes
- For an application server *deployment_directory*/affwebservices/WEB-INF/classes
- For the CA Access Gateway: *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes

*web_agent_home*
Indicates the installed location of the Web Agent.

*deployment_directory*
Indicates the default deployment directory for your application server.

*sps_home*
Indicates the installed location of CA Access Gateway.

Modify the settings as needed. If a value is not specified, the default value for the default locale is used.

The following table shows the settings in the LoggerConfig.properties file.

| LoggerConfig.properties Settings | Description |
|---|---|
| EnableDNSLookup | Instructs the FWS application whether to do a DNS or reverse DNS lookup when processing an incoming SAML request at the consuming site. Select Y or N.<br>When an incoming SAML request is received at a consumer site, FWS logs the details of the request, including the requesting host name. The DNS lookup call collects the host name.<br>The default behavior is to do the DNS lookup. If you select N for this heading, the DNS call is not made and the IP address is logged instead. |
| LoggingOn (required) | Enables log output. |
| LocalFileName (required) | Names the file to use for log output. |
| LogLocalTime | Enables use of local time for log messages. |
| LogRollover | Defines the type of rollover functionality for the error log output files. |
| LogSize | Specifies the maximum file size, in megabytes, when rolling over log files by size. |
| LogCount | Specifies how many log output files to leave when roll-over is enabled. |
| TracingOn | Enables trace log output. |
| TraceFileName | Names the file to use for trace log output. |
| TraceConfig | Specifies the trace configuration file. For more information, see Trace Logging. |
| TraceRollover | Defines the type of rollover functionality for the trace output file. |
| TraceSize | Specifies the maximum file size, in megabytes, when rolling over trace log files by size. |
| TraceCount | Specifies how many trace log output files to leave when roll-over is enabled. |
| TraceFormat | Specifies the trace output file format (default, fixed-width fields, delimited format, XML) |
| TraceDelim | Defines the character to use as a delimiter when using fixed-width fields as the trace format. |

## Deploy FWS on Different Application Servers

If you are using the Web Agent Option Pack, deploy the Federation Web Services (FWS) application into operation.

Configure one of the following application servers to work with FWS:

- Set Up ServletExec to Work with Federation Web Services (see page 894).
- Set Up WebLogic to Work with Federation Web Services (see page 899).
- Set Up WebSphere to Work with Federation Web Services (see page 904).
- Set Up a JBOSS or Tomcat to Work with Federation Web Services (see page 910).

If you are using the CA Access Gateway, Federation Web Services is already deployed on the embedded Tomcat server.

**Note**: Restart your server if a redeployment of Federation Web Services fails with the following error:

```
java.lang.UnsatisfiedLinkError: Native Library 'smerrlog.dll' already l
com.netegrity.smerrlog.SmLogException: Failed to load smerrlog.
```

## Set Up ServletExec to Work with Federation Web Services

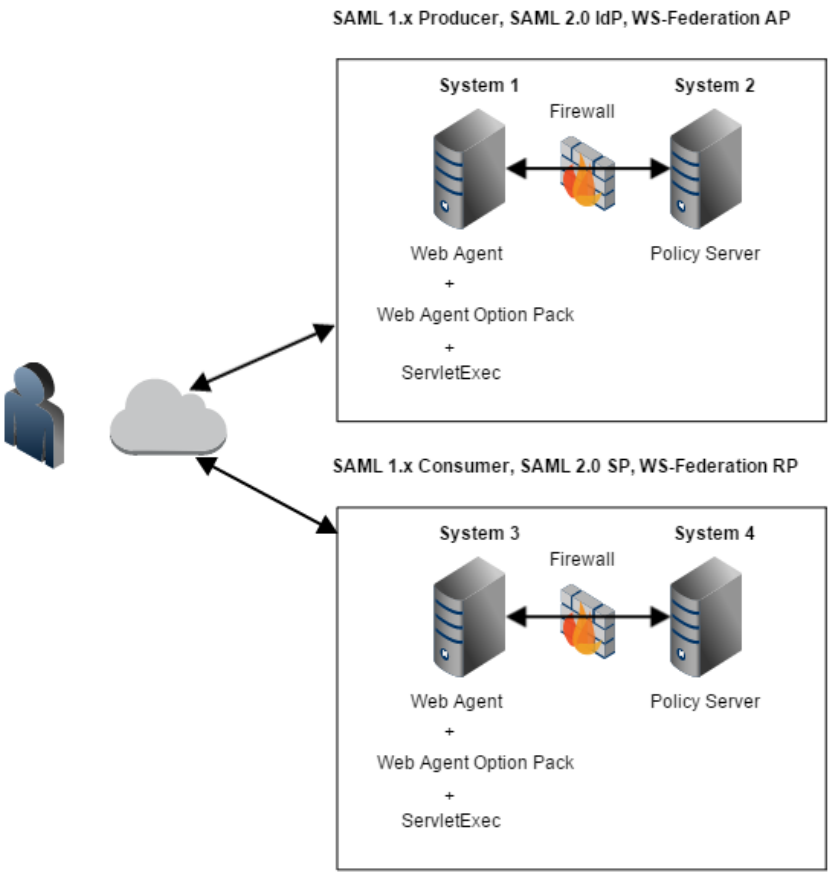
**Contents**



For the Federation Web Services (FWS) application to work with ServletExec, deploy Federation Web Services as a web application for ServletExec. Deploy the FWS application at the asserting and relying party.

**Note:** CA Single Sign-On 12.52 is shipped with a ServletExec license key file named ServletExec_AS_6_license_key.txt. If you do not have this license key, contact CA Technical Support (http://www.ca.com/support). From this license file, copy the license key and enter it in the ServletExec License dialog of the ServletExec Administration Console. For instructions on licensing ServletExec, see ServletExec documentation, available at the website.

The following illustration shows a sample configuration of CA Single Sign-On and ServletExec. ServletExec, the Web Agent Option Pack, and the Web Agent are installed on the same server; however, this setup is not required.




**Important!** Apply the most current hot fixes for ServletExec. Federation Web Services requires the hot fixes to work with ServletExec. To obtain the hot fixes, go to the .

**Follow these steps:**

1. Open the ServletExec Administration Console.
2. Under Web Applications, select manage.
   The Manage Web Applications dialog opens.
3. Click Add a Web Applications.
4. Enter the following information:
   a. Application Name: affwebservices
   b. URL Context Path: /affwebservices/

c. Location: affwebservices_home
Example:
C:\program files\ca\webagent\affwebservices

5. Click Submit.

6. Exit the ServletExec Console.

## Source the Environment Script on a UNIX Operating Environments

After you install the Web Agent Option Pack on a UNIX system, the installation program creates an environment script (ca-wa-opack-env.sh).

Source the environment script so the library path of the application server points to the location of the Web Agent Option Pack /bin directory.

Source the script by entering the following command at the command line:

```
. ./ca-wa-opack-env.sh
```

Setting the correct library path lets the option pack and the web or application server to work together.

After you source the script, the library path is set. The variable name for the library path differs depending on the operating system. Example of several library paths:

- **Solaris/Linux**
  LD_LIBRARY_PATH=/*webagent_option_pack_home*/bin
- **HP-UX**
  SHLIB_PATH=/*webagent_option_pack_home*/bin
- **AIX**
  LIBPATH=/*webagent_option_pack_home*/bin

**Important!** The application server startup script can reset the library path. Ensure that the path to the Web Agent Option Pack is the first entry in the path.

The path to the Web Agent Option Pack environment script points to one of the following locations:

- The installation directory of the web agent option pack. The default location is: */webagent_option_pack_home/bin*.
- The installation directory of the web agent.
  If you install the option pack on the same system as the web agent, the script resides in the web agent directory. For any UNIX installation, the default location is */web_agent_home/bin*.

## Modify the FWS Properties File for a ServletExec Deployment

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. For deploying FWS, set only the parameter that specifies the location of the WebAgent.conf file.

**To configure the AffWebServices.properties file**

1. Navigate to the AffWebServices.properties file. For ServletExec, go to *web_agent_home* /affwebservices/WEB-INF/classes.
2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file at each partner site.
   - Windows example:
     C:\\Program Files\\ca\\webagent\\bin\\IIS\\WebAgent.conf

     **Note:** Federation Web Services is a Java component, so the Windows paths must contain double backslashes.
   - UNIX example:
     *server_home*/servers/https-hostname/config/WebAgent.conf
   - Windows example for the CA Access Gateway
     *sps_home*\\proxy-engine\\conf\\defaultagent\\WebAgent.conf
   - UNIX example for the CA Access Gateway
     *sps_home*/proxy-engine/conf/defaultagent/WebAgent.conf
3. Repeat this procedure for each application server where the Web Agent Option Pack is installed.
4. Accept the default values for the rest of the settings.

## Enable ServletExec to Write to the IIS File System

The IIS server user account must have proper rights for IIS to allow a plug-in to write to its file system. For ServletExec to write to the federation log files, the anonymous user account that is associated with ServletExec must have permissions to write to the file system.

**Follow these steps:**

1. Open the IIS Internet Information Services Manager on the system where ServletExec is installed.
2. Navigate to Web Sites, Default Web Site.
   The set of applications is displayed in the right pane.
3. Select ServletExec and right-click Properties.

4. Select the Directory Security tab in the Properties dialog.

5. Click Edit in the Authentication and access control section.
   The Authentication Methods dialog opens.

6. Set the controls as follows.

   a. Select Enable Anonymous Access.
      For anonymous access, enter a name and password of a user account that has the permissions to right to the Windows file system. To grant this right to a user account, see Windows documentation. For example, you can use the IUSR Internet Guest account for anonymous access.

   b. Clear Basic authentication.

   c. Clear Integrated Windows authentication.

7. If prompted, apply the security changes to all child components of the web server.

8. Restart the web server.

The user account that is associated with ServletExec can now write to the IIS file system.

**Follow these steps:**

1. Open Control Panel, Administrative Tools, Local Security Policy, Local Policies, User Rights Assignment.
   The Local Security Settings dialog displays.

2. Double-click Act as part of the operating system.
   The Act as part of the operating system Properties dialog opens.

3. Add the anonymous user account to the Local Security Setting dialog.

4. Click OK.

5. Exit from the control panel.

Optionally, we strongly recommend that you look at the Agent Configuration Object for the Web Agent protecting the IIS Web Server. This object verifies that the SetRemoteUser parameter is set to yes to preventing any anonymous user from writing to the file system.

## Ensure the IIS Default Web Site Exists

The Web Agent requires the IIS Web Server to have a Default Web Site for proper installation. The Default Web Site is automatically installed with the IIS Web Server. If this website does not exist, install the CA Single Sign-On virtual directories to a different website on IIS. To install the CA Single Sign-On virtual directories to a different website on IIS, edit the Metabase.

A technical note on the site describes the Technical Support site (http://www.ca.com/support) changes that are needed. To find the note:

1. Go to the main Support page.

2. Select Literature, Tech Notes.

3. Select the document titled METABASE -3 Error.
   The documents are listed in alphabetical order.

## Set Up WebLogic to Work with Federation Web Services

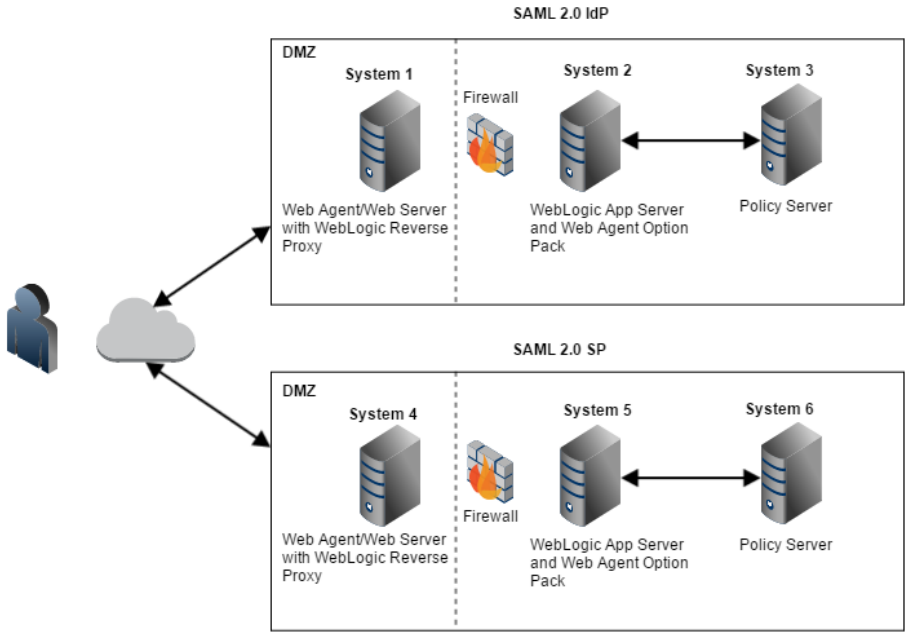
**Contents**



To enable Federation Web Services (FWS) for a CA Single Sign-On/WebLogic configuration, deploy the FWS application.

**Note:** For a list of supported version of WebLogic, see the CA Single Sign-On 12.52 Platform Support Matrix on the Technical Support site.

The following illustration shows a CA Single Sign-On and WebLogic sample configuration. The illustration provides an example of how to use FWS in a sample federated environment.




In this environment, deploy the FWS application on System 2 and System 5.

**Important!** Complete the deployment procedure for the Web Agent at the asserting party and the relying party.

After installing the software components on the systems in the illustration, deploy the FWS application. Deploy the application on System 2 for the asserting party and on System 5 for the relying party.

**To deply the FWS application**

1. Set the LD_LIBRARY_PATH Variable
2. Create a SmHost.conf File
3. Create a WebAgent.conf File
4. Modify the AffWebServices.properties File
5. Configure the WebLogic Reverse Proxy Plug-in.
6. Deploy the FWS Application on WebLogic.

**Important!** For the FWS application to work with WebLogic Server, review the weblogic.xml file in the WEB-INF directory. Verify that the prefer-web-inf-classes parameter in the weblogic.xml file is set to true.

For instructions on reviewing the weblogic.xml file, go to Deploy the FWS Application on WebLogic (see page 903).

## Source the Environment Script on a UNIX Operating Environments

After you install the Web Agent Option Pack on a UNIX system, the installation program creates an environment script (ca-wa-opack-env.sh).

Source the environment script so the library path of the application server points to the location of the Web Agent Option Pack /bin directory.

Source the script by entering the following command at the command line:

```
. ./ca-wa-opack-env.sh
```

Setting the correct library path lets the option pack and the web or application server to work together.

After you source the script, the library path is set. The variable name for the library path differs depending on the operating system. Example of several library paths:

- **Solaris/Linux**
  LD_LIBRARY_PATH=/*webagent_option_pack_home*/bin

- **HP-UX**
  SHLIB_PATH=/*webagent_option_pack_home*/bin

- **AIX**
  LIBPATH=/*webagent_option_pack_home*/bin

**Important!** The application server startup script can reset the library path. Ensure that the path to the Web Agent Option Pack is the first entry in the path.

The path to the Web Agent Option Pack environment script points to one of the following locations:

- The installation directory of the web agent option pack. The default location is: */webagent_option_pack_home/bin*.

- The installation directory of the web agent.
  If you install the option pack on the same system as the web agent, the script resides in the web agent directory. For any UNIX installation, the default location is */web_agent_home/bin*.

## Create an SmHost.conf File

The FWS application requires an SmHost.conf file. However, the Web Agent Option Pack does not install this file, so you must create it.

**To create an SmHost.conf**

1. Go to the directory /*webagent_option_pack_home*/bin

2. Run the smreghost.exe.

3. Put the SmHost.conf file in the following directory on Systems 2 and 5:
   /*webagent_option_pack_home*/config

## Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file. However, the Web Agent Option Pack does not install this file, so you must create it.

**To create a WebAgent.conf file**

1. Copy the WebAgent.conf file from System 1 to the following directory on System 2 and System 5:
   /*webagent_option_pack_home*/config

   - ***webagent_option_pack_home***
     Defines the installed location of the Web Agent Option Pack on System 2 or System 5.

2. Modify the WebAgent.conf file by:

   a. Setting the EnableWebAgent parameter to YES.

b. Modifying other configuration parameters to suit FWS.

The following sample shows a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname="agent_name, IP_address"
HostConfigFile="/webagent_option_pack/config/SmHost.conf"
AgentConfigObject="agent_config_object_name"
EnableWebAgent="YES"
```

## Modify the FWS Properties File

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. For deploying FWS, set only the parameter that specifies the location of the WebAgent.conf file.

**Follow these steps:**

1. Navigate to the AffWebServices.properties file. Locate this file in the following directory:
   *web_agent_optionpack_home*/affwebservices/WEB-INF/classes

2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file at each partner site.

   - Windows example:
     C:\\Program Files\\CA\\webagent_optionpack\\config\\WebAgent.conf

     **Note:** Federation Web Services is a Java component, so the Windows paths must contain double backslashes.

   - UNIX example:
     *web_agent_optionpack_home*/config/WebAgent.conf

   - Windows example for the CA Access Gateway
     *sps_home*\\proxy-engine\\conf\\defaultagent\\WebAgent.conf

   - UNIX example for the CA Access Gateway
     *sps_home*/proxy-engine/conf/defaultagent/WebAgent.conf

3. Repeat this procedure for each application server where the Web Agent Option Pack is installed.

4. Accept the default values for the rest of the settings in the properties file.

## Configure the WebLogic Reverse Proxy Plug-in

To set up the WebLogic Reverse Proxy plug-in:

1. On System 1, configure the WebLogic reverse proxy plug-in on the Apache Web Server. For more information, see WebLogic documentation.

2. Add the following aliases to the configuration file of the web server.
   This example uses the Apache httpd.conf file.

```
<IfModule mod_weblogic.c>
WebLogicHost <WebLogic_Machine_IP Address>
WebLogicPort <WebLogic_Machine_Port_Number>
</IfModule>

<Location /affwebservices>
SetHandler weblogic-handler
Debug ALL
</Location>
```

## Deploy the FWS Application on WebLogic

Deploy the FWS application on System 2 and System 5.

**Important!** For the FWS application to work with WebLogic Server, review the weblogic.xml file in the WEB-INF directory. Verify that the prefer-web-inf-classes parameter is set to true.

The weblogic.xml file is located in the directory webagent\affwebservices\WEB-INF.

The following code excerpt shows how to set the prefer-web-inf-classes parameter:

```
<weblogic-web-app>
 <container-descriptor>
  <prefer-web-inf-classes>true</prefer-web-inf-classes>
 </container-descriptor>
</weblogic-web-app>
```

In addition, verify that the precompile parameter is set to true, as listed following:

```
<jsp-descriptor>
   <precompile>true</precompile>
<jsp-descriptor>
```

**Follow these steps:**

1. Use the WebLogic Server Console and deploy FWS. The FWS application is installed in:
   */webagent_option_pack_home*/affwebservices/
   For more information about deploying a web application, see the WebLogic documentation.

2. Test that the FWS application is working. Open a web browser and enter:
   http://*fqhn*:*port_number*/affwebservices/assertionretriever

   - ***fqhn***
     Defines the fully qualified host name.

   - ***port_number***
     Defines the port number of the server where the Federation Web Services application is installed.

For example:
http://myhost.ca.com:81/affwebservices/assertionretriever
If Federation Web Services is operating correctly, you see the following message:

```
Assertion Retrieval Service has been successfully initialized.
The requested servlet accepts only HTTP POST requests.
```

This message indicates that Federation Web Services is listening for data activity. The FWS application is now deployed for the WebLogic server.
If Federation Web Services is not operating correctly, a message that the Assertion Retrieval Service has failed displays. If the service fails, review the Federation Web Services log.

**Note:** For instructions on enabling trace logging for the FWS application, see Trace Logging.

## Set Up WebSphere to Work with Federation Web Services

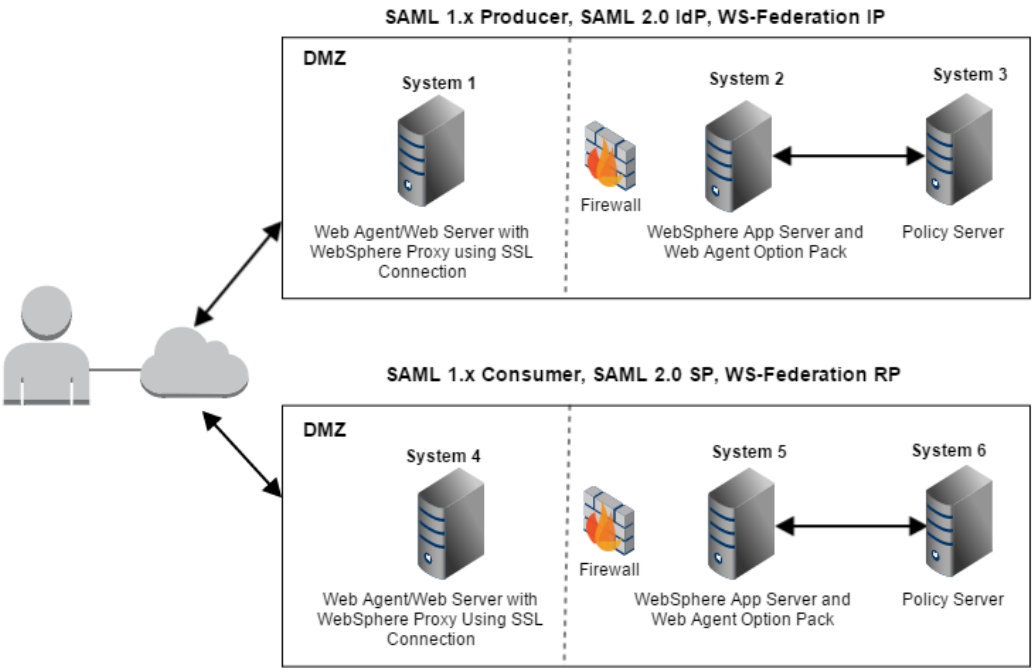
**Contents**



To enable CA Single Sign-On federation on a WebSphere Application Server (WAS), deploy the FWS application.

On Systems 2 and 5, deploy FWS. These systems must also have WAS and the associated WAS Fix Pack, if applicable. On Systems 1 and 4, install the Web Agent and the WAS Proxy Plug-in. Enable SSL between the proxy and the WAS.

The following illustration shows a WebSphere sample configuration:




**Prerequisites**

- Install FWS on systems that have the WebSphere Application Server installed.
- Complete the deployment procedure for the Web Agent at the asserting party and the relying party.
- Change the JDK Source Level on IBM WebSphere
  The redirect.jsp page included with the Web Agent Option Pack uses Java/JDK version 1.5. The IBM WebSphere application server uses a lower Java/JDK version to compile JSP pages. When affwebservices is deployed on IBM WebSphere, the redirect.jsp page can have a problem compiling because of the different JDK source level.

  Set the Java/JDK version to 1.5 to complie JSP pages. Edit the im-web-ext.xmi file and set the **jdkSourceLevel** attribute to 15, as follows:

```
<jspAttributes xmi:id="JSPAttribute_1403542495646" name="jdkSourceLevel" value="15"
/>
```

After installing the software components on the systems in the illustration, deploy FWS on System 2 and System 5 by following these steps:

1. Set the WebSphere LD_LIBRARY_PATH variable.
2. Create a SmHost.conf file.
3. Create a WebAgent.conf file.
4. Modify the AffWebServices.properties file.
5. Copy option pack library files to WebSphere.

6. Deploy a Federation Web Services WAR File in WebSphere.

## Source the Environment Script on a UNIX Operating Environments

After you install the Web Agent Option Pack on a UNIX system, the installation program creates an environment script (ca-wa-opack-env.sh).

Source the environment script so the library path of the application server points to the location of the Web Agent Option Pack /bin directory.

Source the script by entering the following command at the command line:

```
. ./ca-wa-opack-env.sh
```

Setting the correct library path lets the option pack and the web or application server to work together.

After you source the script, the library path is set. The variable name for the library path differs depending on the operating system. Example of several library paths:

- **Solaris/Linux**
  LD_LIBRARY_PATH=/*webagent_option_pack_home*/bin
- **HP-UX**
  SHLIB_PATH=/*webagent_option_pack_home*/bin
- **AIX**
  LIBPATH=/*webagent_option_pack_home*/bin

**Important!** The application server startup script can reset the library path. Ensure that the path to the Web Agent Option Pack is the first entry in the path.

The path to the Web Agent Option Pack environment script points to one of the following locations:

- The installation directory of the web agent option pack. The default location is: */webagent_option_pack_home/bin*.
- The installation directory of the web agent.
  If you install the option pack on the same system as the web agent, the script resides in the web agent directory. For any UNIX installation, the default location is */web_agent_home/bin*.

## Create an SmHost.conf File

The FWS application requires the SmHost.conf file. However, the Web Agent Option Pack does not install this file, so you must create it.

**To create an SmHost.conf file**

1. Create an SmHost.conf file by running smreghost.exe, which is located in the following directory:
   */webagent_option_pack_home*/bin

2. Put the SmHost.conf file in the following directory on System 2 and System 5:
   */webagent_option_pack_home*/config

## Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file; however, the Web Agent Option Pack does not install this file so you must create it.

**To create a WebAgent.conf file**

1. Copy the WebAgent.conf file from System 1 to the following directory on System 2 and System 5:
   */webagent_option_pack_home*/config
   where,

   - ***webagent_option_pack_home***
     Defines the installed location of the Web Agent Option Pack on System 2 and System 5.

2. Modify the WebAgent.conf file by:

   a. Setting the EnableWebAgent parameter to YES.

   b. Modifying any other configuration parameters to suit the environment for the FWS application.

The following sample shows a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname="<agent_name>, <IP_address>"
HostConfigFile="/<webagent_option_pack>/config/SmHost.conf"
AgentConfigObject="<agent_config_object_name>"
EnableWebAgent="YES"
```

## Modify the FWS Properties File

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. For deploying FWS, set only the parameter that specifies the location of the WebAgent.conf file.

**Follow these steps:**

1. Navigate to the AffWebServices.properties file. Locate this file in the following directory:
   *web_agent_optionpack_home*/affwebservices/WEB-INF/classes

2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file at each partner site.

   - Windows example:
     C:\\Program Files\\CA\\webagent_optionpack\\config\\WebAgent.conf

> **Note:** Federation Web Services is a Java component, so the Windows paths must contain double backslashes.

- UNIX example:
  *web_agent_optionpack_home*/config/WebAgent.conf
- Windows example for the CA Access Gateway
  *sps_home*\\proxy-engine\\conf\\defaultagent\\WebAgent.conf
- UNIX example for the CA Access Gateway
  *sps_home*/proxy-engine/conf/defaultagent/WebAgent.conf

3. Repeat this procedure for each application server where the Web Agent Option Pack is installed.
4. Accept the default values for the rest of the settings in the properties file.

## Copy Web Agent Option Pack Libraries to WebSphere

Copy the Web Agent Option Pack library files which reside on System 2 and System 5.

**Follow these steps:**

1. Go to the directory *webagent_optionpack_home*\bin.
2. From the bin folder, copy the following files for your operating platform and place them in the directory *WebSphere_home*\AppServer\bin.

| Windows | UNIX (except AIX) | AIX |
|---|---|---|
| ▪ icuuc49<br>▪ icudt49<br>▪ icuin49<br>▪ smcommonutil.dll<br>▪ smerrlog.dll<br>▪ smfedclientcomponent.dll<br>▪ smjavaagentapi.dll<br>▪ smi18n<br>▪ SmXlate | ▪ All files that begin with the prefix **libicu**<br>such as, libicuuc49, lib icudata 49<br>▪ libsmcommonutil.dll<br>▪ libsmerrlog.dll<br>▪ libsmfedclientcomponent.dll<br>▪ libsmjavaagentapi.dll<br>▪ libsmi18n<br>▪ libSmXlate | ▪ All files that begin with the prefix **libsicu**<br>such as, libsicuuc49, libs icudata49<br>▪ libssmcommonutil.dll<br>▪ libssmerrlog.dll<br>▪ libssmfedclientcomponent.dll<br>▪ libssmjavaagentapi.dll<br>▪ libssmi18n<br>▪ libsSmXlate |

## Deploy a Federation Web Services WAR File in WebSphere

**To deploy the FWS WAR file**

1.

a. Create a WAR file of the Federation Web Services application. The application is installed in:
*webagent_option_pack*\affwebservices\
For more information about creating a WAR file, see WebSphere documentation.

b. Deploy the WAR file using WebSphere Administrator Console.
For more information, see WebSphere documentation.

> **Important!** If you make subsequent changes to any of the properties files in the affwebservices directory, recreate a WAR file and redeploy this file in the application server.

c. From the WebSphere Administrator Console, go to Applications, Enterprise Applications.

d. Select the name of the web services WAR file, such as affwebservices_war.

e. On the Configuration tab:

   i. Set the Classloader Mode.
   There are two possible modes for class loading:

   - Classes loaded with the parent class loader first (default)
   - Classes loaded with the local class loader first

   The mode you select is implementation-dependent. In releases before 7.0, these modes were named PARENT_FIRST and PARENT_LAST. See the WebSphere documentation for further information.

   ii. Set WAR Classloader Policy to Application.

   iii. Save the settings.

f. Test that the Federation Web Services application is working by opening a web browser and entering:
http://*fqhn*:*port_number*/affwebservices/assertionretriever
where,

- ***fqhn***
  Defines the fully qualified host name.
- ***port_number***
  Defines the port number of the server where the Federation Web Services application is installed.

For example:
http://myhost.ca.com:81/affwebservices/assertionretriever
If Federation Web Services is operating correctly, the following message appears:

```
Assertion Retrieval Service has been successfully initialized.
The requested servlet accepts only HTTP POST requests.
```

This message indicates that Federation Web Services is listening for data activity. When the Federation Web Services is not operating correctly, a message states that the Assertion Retrieval Service has failed. If the Assertion Retrieval Service fails, verify the Federation Web Services log.

**Note:** For more information about enabling trace logging for the FWS application, see Trace Logging.

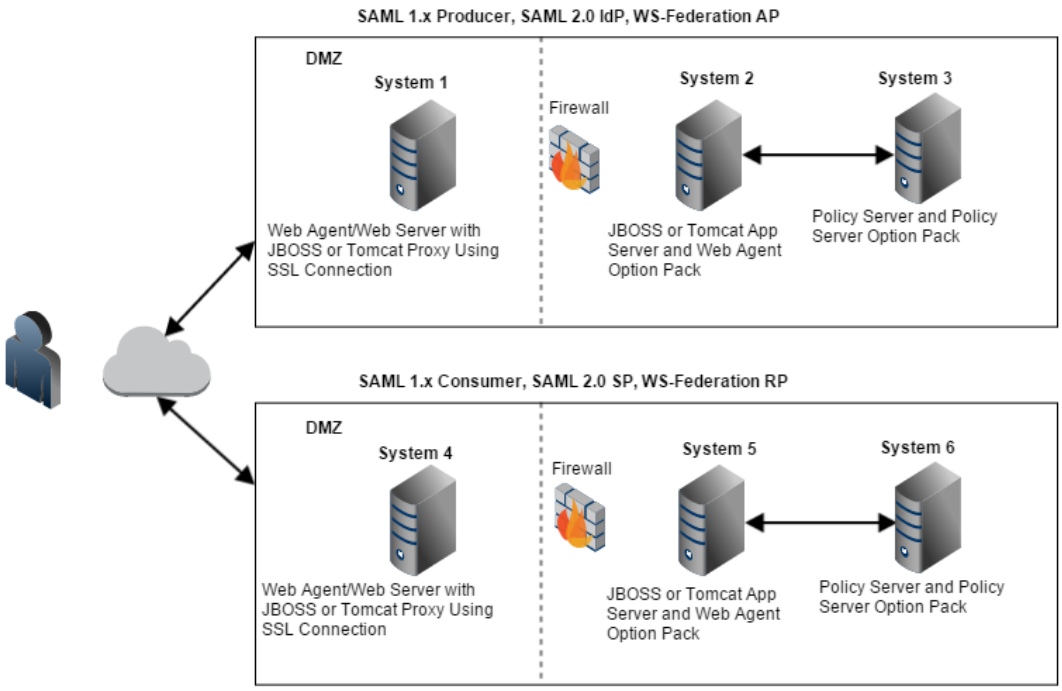
## Set Up JBOSS or Tomcat to Work with Federation Web Services

**Contents**



To use a JBoss or Tomcat Application Server in a CA Single Sign-On federated environment, deploy the FWS application on the application server.

The following illustration shows the deployment with JBOSS or Tomcat. On Systems 1 and 4, the Web Agent is installed with the JBOSS or Tomcat Connector for proxy support. SSL is enabled between the proxy and the application server. On Systems 2 and 5, FWS is deployed with the application server by way of the Web Agent Option Pack.




The process for deploying FWS is as follows:

1. (UNIX) Source the environment script on UNIX operating environments.
2. Create an SmHost.conf file.
3. Create a WebAgent.conf file.
4. Modify the AffWebServices properties file.
5. Deploy the FWS WAR file in the application server.

The following sections detail each step in the process.

## (UNIX) Source the Environment Script on a UNIX Operating Environments

After you install the Web Agent Option Pack on a UNIX system, the installation program creates an environment script (ca-wa-opack-env.sh).

Source the environment script so the library path of the application server points to the location of the Web Agent Option Pack /bin directory.

Source the script by entering the following command at the command line:

```
. ./ca-wa-opack-env.sh
```

Setting the correct library path lets the option pack and the web or application server to work together.

After you source the script, the library path is set. The variable name for the library path differs depending on the operating system. Example of several library paths:

- **Solaris/Linux**
  LD_LIBRARY_PATH=/*webagent_option_pack_home*/bin
- **HP-UX**
  SHLIB_PATH=/*webagent_option_pack_home*/bin
- **AIX**
  LIBPATH=/*webagent_option_pack_home*/bin

**Important!** The application server startup script can reset the library path. Ensure that the path to the Web Agent Option Pack is the first entry in the path.

The path to the Web Agent Option Pack environment script points to one of the following locations:

- The installation directory of the web agent option pack. The default location is: */webagent_option_pack_home/bin*.
- The installation directory of the web agent.
  If you install the option pack on the same system as the web agent, the script resides in the web agent directory. For any UNIX installation, the default location is */web_agent_home/bin*.

## Create an SmHost.conf File

The FWS application requires the SmHost.conf file. However, the Web Agent Option Pack does not install this file, so you must create it.

1. Create an SmHost.conf file by running smreghost.exe, which is located in the following directory:
   Windows: *webagent_option_pack*\bin
   UNIX: /*webagent_option_pack*/bin
2. Put the SmHost.conf file in the following directory on System 2 and System 5:
   Windows: *webagent_option_pack*\config
   UNIX: /*webagent_option_pack*/config

## Create a WebAgent.conf File

The FWS application requires the WebAgent.conf file. However, the Web Agent Option Pack does not install this file, so you must create it.

1. Copy the WebAgent.conf file from System 1 to the following directory on System 2 and System 5:
   Windows: *webagent_option_pack*\config
   UNIX: /*webagent_option_pack*/config
   where,

- ***webagent_option_pack***
  Defines the installed location of the Web Agent Option Pack on System 2 and System 5.

2. Modify the WebAgent.conf file as follows:

   a. Set the EnableWebAgent parameter to YES.

   b. Modify any other configuration parameters to suit the environment for the FWS application.

   The following sample shows a WebAgent.conf file for the FWS application:

```
# WebAgent.conf - configuration file for the Federation Web Services Application
#agentname="agent_name, IP_address"
HostConfigFile="/webagent_option_pack_home/config/SmHost.conf"
AgentConfigObject="agent_config_object_name"
EnableWebAgent="YES"
```

## Modify the FWS Properties File

The AffWebServices.properties file contains all the initialization parameters for Federation Web Services. For deploying FWS, set only the parameter that specifies the location of the WebAgent.conf file.

**Follow these steps:**

1. Navigate to the AffWebServices.properties file. Locate this file in the following directory:
   *web_agent_optionpack_home*/affwebservices/WEB-INF/classes

2. Set the AgentConfigLocation parameter to the location of the WebAgent.conf file at each partner site.

   - Windows example:
     C:\\Program Files\\CA\\webagent_optionpack\\config\\WebAgent.conf

     **Note:** Federation Web Services is a Java component, so the Windows paths must contain double backslashes.

   - UNIX example:
     *web_agent_optionpack_home*/config/WebAgent.conf

   - Windows example for the CA Access Gateway
     *sps_home*\\proxy-engine\\conf\\defaultagent\\WebAgent.conf

   - UNIX example for the CA Access Gateway
     *sps_home*/proxy-engine/conf/defaultagent/WebAgent.conf

3. Repeat this procedure for each application server where the Web Agent Option Pack is installed.

4. Accept the default values for the rest of the settings in the properties file.

## Complete JBoss Deployment Prerequisites (Optional)

For JBoss 6.1, there are two prerequisites before you can deploy FWS. The steps are required because the affwebservices war file fails to deploy by default.

To deploy the affwebservices war file:

1. Update the affwebservices deployment descriptor file.
2. Create a module.xml file.

If you are not using JBoss 6.1, go to Deploy an FWS WAR File (JBoss or Tomcat). (see page 915)

## Update the Affwebservices Deployment Descriptor File

Edit the affwebservices deployment descriptor to add a few <context-param> entries.

**Follow these steps:**

1. Open the affwebservices deployment descriptor file (*webagent_option_pack*/affwebservices /WEB-INF/web.xml) in a text editor.
2. Add the following lines after the <web-app> tag and before the <servlet> tag:

```
<context-param>
<param-name>resteasy.scan</param-name>
<param-value>false</param-value>
</context-param>
<context-param>
<param-name>resteasy.scan.resources</param-name>
<param-value>false</param-value>
</context-param>
<context-param>
<param-name>resteasy.scan.providers</param-name>
<param-value>false</param-value>
</context-param>
```

3. Save and exit the text editor.

## Create a module.xml File

To deploy the war file, create a directory in the JBoss container, associate the jars files, and create a module.xml file that describes the jar files.

**Follow these steps:**

1. Create a directory structure as follows under the <JBOSS-_HOME>\modules location:

   ```
   com\rsa\cryptoj\main
   ```

2. Copy the cryptoj.jar, certj.jar, and sslj.jar files from the following location:

   ```
   <NETE_WA_ROOT>\affwebservices\WEB-INF\lib\
   ```

   to

   ```
   <JBOSS-HOME>\modules\com\rsa\cryptoj\main\
   ```

3. Create a module.xml file in the following location:

   ```
   <JBOSS-HOME>\modules\com\rsa\cryptoj\main
   ```

4. Add the following to the xml file:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <module xmlns="urn:jboss:module:1.0" name="com.rsa.cryptoj">
       <resources>
           <resource-root path="cryptoj.jar"/>
           <resource-root path="certj.jar"/>
           <resource-root path="sslj.jar"/>
       </resources>
       <dependencies>
           <module name="sun.jdk"/>
           <module name="javax.api"/>
       </dependencies>
   </module>
   ```

5. Restart the JBoss server.

You can deploy the affwebservices war file in the JBoss server.

## Deploy the FWS Application on JBoss or Tomcat

**Follow these steps:**

1. Open a command window and navigate to the affwebservices directory, which is located in:

   ```
   /webagent_option_pack/affwebservices/.
   ```

2. Create a WAR file by entering the command:

   ```
   jar cvf affwebservices.war *
   ```

   For more information about deploying a Web application, see the documentation for your application server.

3. Copy the affwebservices.war file to the appropriate server location:

   - **JBOSS**
     *JBOSS_home*/server/default/deploy/
     *JBOSS_home* is the installed location of the JBOSS application server.

     **Note:** For JBoss EAP 6.1, use the admin console to deploy the affwebservices.war file.

     **Important!** For JBOSS, deploy affwebservices in an exploded state. Refer to https://access.redhat.com/knowledge/solutions/34813.

   - **Tomcat**
     *Tomcat_home*/webapps
     *Tomcat_home* is the installed location of the Tomcat application server.

4. Restart the application server.

5. After the server has restarted, access the JBOSS or Tomcat Administrative Console. All the services that affwebservices supports appear on the main Console page.

6. Test that the FWS application is working by opening a web browser and entering the following URL:
   http://*fqhn*:*port_number*/affwebservices/assertionretriever

   - **fqhn**
     Represents the fully qualified host name and

   - **port_number**
     Specifies the port number of the server where the Federation Web Services application is installed.

   For example:
   http://myhost.ca.com:81/affwebservices/assertionretriever
   If FWS is operating correctly, the following message is displayed:

   ```
   Assertion Retrieval Service has been successfully initialized.
   The requested servlet accepts only HTTP POST requests.
   ```

   This message indicates that FWS is listening for data activity. The FWS application is now deployed for the application server.
   When FWS is not operating correctly, a message states that the Assertion Retrieval Service has failed. If there is a failure, review the FWS log.

   **Note:** For more information about enabling trace logging for the FWS application, see Trace Logging.

# Unattended Mode Installation

**Contents**



## How to Run an Unattended Mode Installation

After the Web Agent Option Pack is installed on one system, you can automate installations on other Web or application servers using the Web Agent Option Pack's unattended mode installation. An unattended mode installation lets you install or uninstall the Web Agent Option Pack without any user intervention.

## Prepare an Unattended Mode Installation

An unattended mode installation uses the ca-wa-opack-installer.properties file to propagate the Option Pack installation set-up across all servers in your network. You can define the installation parameters in the properties file then copy the properties file and the Web Agent Option Pack executable file to any applicable server in your network. After the files are copied, you can run an unattended installation.

**To prepare an unattended mode installation**

1. Run an initial installation of the Web Agent Option Pack in GUI or Console mode.
   This installation will install the ca-wa-opack-installer.properties file.

2. Open the ca-wa-opack-installer.properties file, and if needed, modify the settings.
   The properties file is in the directory *web_agent_opack_home*/install_config_info.

   - **USER_INSTALL_DIR**
     Specifies the Web Agent Option Pack's installation location.

   - **USER_REQUESTED_RESTART**
     Specifies restarting the machine after installation.

   **Note:** These default values were saved in the properties file during the initial installation.

3. Save the properties file.

## Run an Unattended Mode Installation

You can run an unattended mode installation.

**Follow these steps:**

1. Be sure that you have completed the preparation steps.

2. Copy the following files to a local directory on the system where you want to install the option pack.

   - Web Agent Option Pack executable or binary

     - ca-wa-opack-*version-windows_platform*.exe

     - ca-wa-opack-*version-operating_system*.bin

   - ca-wa-opack-installer.properties

3. Open a console window and navigate to the location where you copied the files.

4. Execute the following command:
   *webagent_option_pack_executable* -f *properties_file* -i silent
   **Windows example:**
   ca-wa-opack-12.52-win32.exe -f ca-wa-opack-installer.properties -i silent
   **Solaris example:**
   ./ca-wa-opack-12.52-sol.bin -f ca-wa-opack-installer.properties -i silent
   These examples assume that you are running the installation from the directory containing the executable and properties files. If you are not running the installation from this directory, specify the full path to these files. If there are spaces in the directory path, enclose the entire path in quotation marks.
   The progress of the unattended installation is displayed. When the installation is complete, the command prompt is redisplayed.

5. Determine that the installation completed successfully by viewing the log file CA_CA Single Sign-on_Option_Pack_12.52_for_Web_Agent_ InstallLog.log. This file is located in the directory *web_agent_opack_home*/install_config_info.

### Stop an Unattended Mode Installation in Progress

**Follow these steps:**

- **Windows**
  Open the Windows Task Manager, and stop the following two processes:
  - ca-wa-opack-12.52-win32.exe
  - wa_option_pack.exe
- **UNIX**
  Type Ctrl+C.

# Uninstall the Web Agent Option Pack

**Contents**



## Uninstall the Web Agent Option Pack from Windows Systems

Uninstall the Web Agent Option Pack to remove it from your Windows system.

**Follow these steps:**

1. Stop the web server.
2. Exit any applications that are running.
3. Open the Windows Control Panel.
4. Double-click Add or Remove Programs.

5. Select CA CA Single Sign-On Option Pack for Web Agent.
6. Click Remove.
7. Confirm the action.
8. Click Uninstall.
9. Stop and restart the web server.

## Uninstall the Web Agent Option Pack from UNIX Systems

You can uninstall the Web Agent Option Pack from UNIX systems in GUI or console mode. To uninstall in console mode, execute the Option Pack binary file with the command argument "-i console".

**Follow these steps:**

1. Stop the Web server, and exit any applications that are running.
2. Open a console window.
3. Add the location of the JDK to the PATH environment variable, as follows:

   ```
   PATH=/jdk_home/bin:${PATH}
   export PATH
   ```

   *jdk_home* is the location of the JDK. This step eliminates the error message which states that the Java virtual machine cannot be found.
4. Navigate to the installed location of the Web Agent.
   **Example:** /opt/ca/webagent/
5. At the prompt, type one of the following commands:
   **GUI Mode:** ./ca-wa-opack-uninstall.sh
   **Console Mode:** ./ca-wa-opack-uninstall.sh -i console
6. Review the dialog box that prompts you to confirm the removal of the Option Pack; then click Uninstall.

The Web Agent Option Pack is now removed from your system.

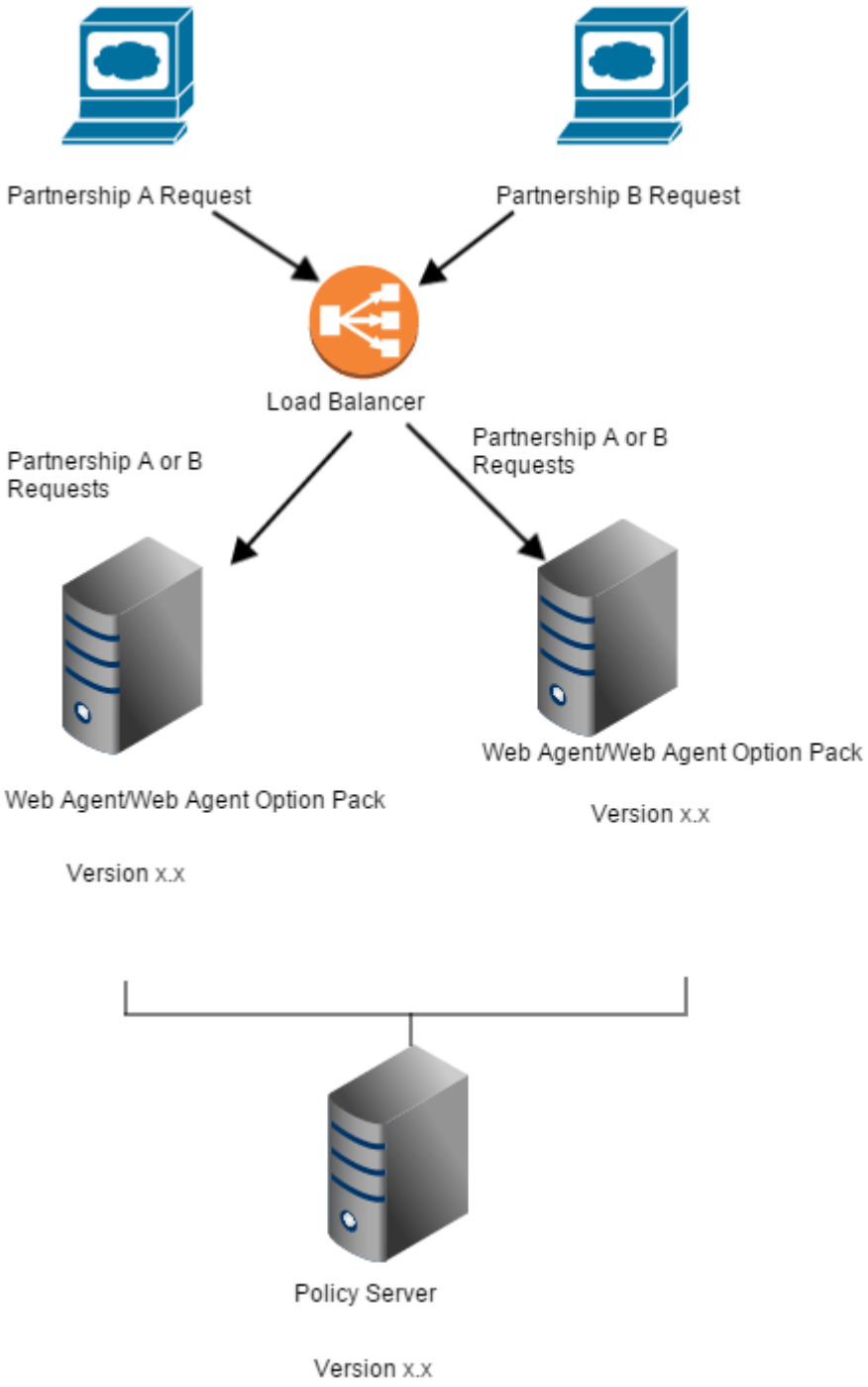
# Upgrade the Web Agent Option Pack

**Contents**

## Mixed-Version Upgrade Considerations

To simplify migrations and upgrades, the Web Agent Option Pack can operate with Policy Servers of different, compatible versions. This mixed-version support allows you to keep Web Agents at their current version while you upgrade Policy Servers and other Web Agents. However, mixed-version environments are not recommended for permanent use because they limit the federation functionality to that of the lesser of the two components.

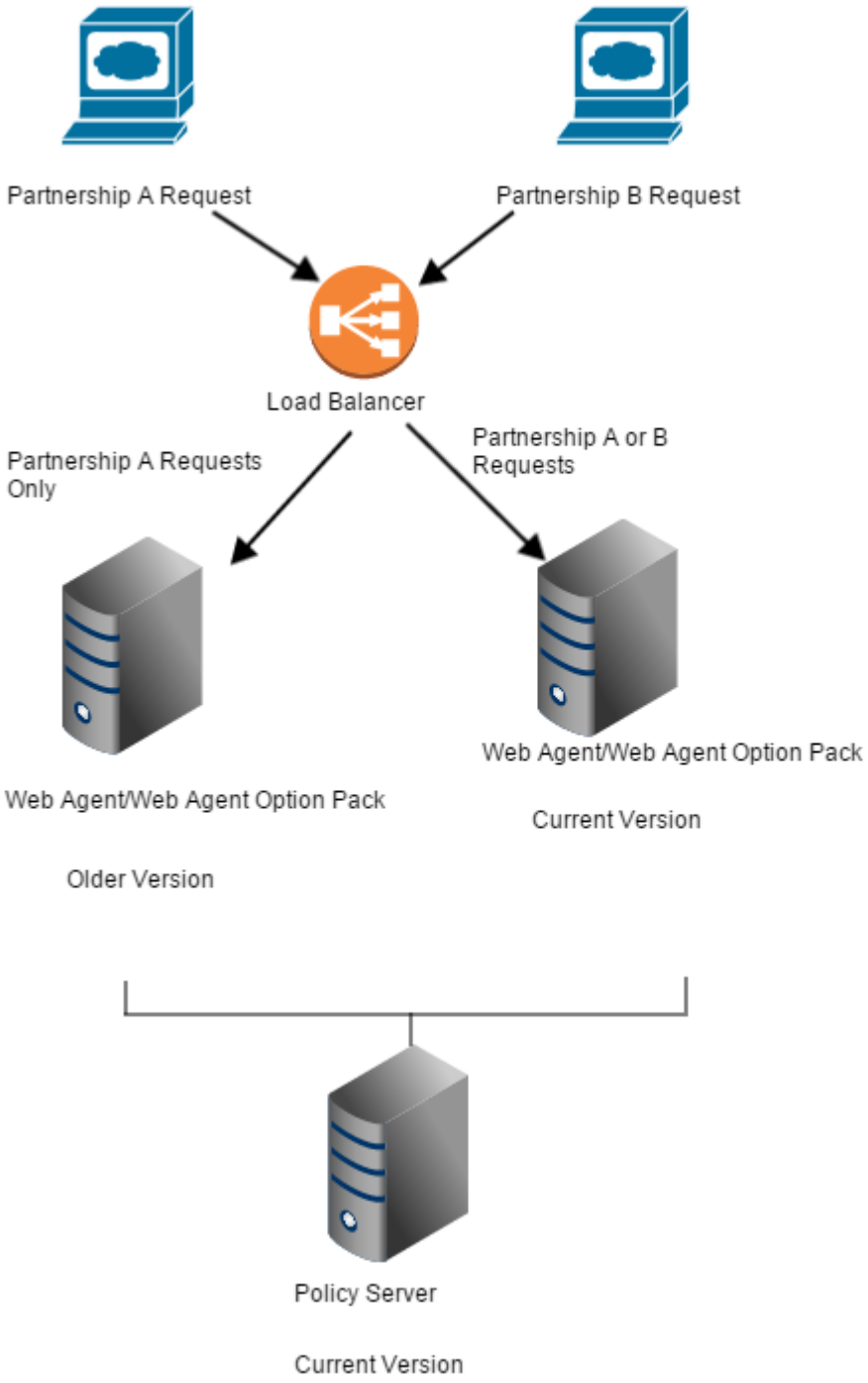
Version-dependent functional limitations exist when upgrading environments that use hardware load balancers for distributing requests between web servers. Consider a deployment where two Web Agents with Web Agent Option Packs and a Policy Server are at the same older version, as shown in the following diagram:




During a rolling upgrade, the administrator upgrades the Policy Server and only one Web Agent and Option Pack to 12.52. The upgraded agent and option pack supports SAML 2.0 user consent at the IdP. Partnership B is reconfigured to require user consent at the IdP and Partnership A security configuration remains unchanged.

Partnership A functionality continues to work regardless of the option pack to which it is routed because it is not using new functionality. However, Partnership B requests (which require user consent at the IdP) fail whenever the load balancer directs the request to the older version option pack.

To prevent Partnership B request failures until both web agents are upgraded, the administrator must configure the load balancer to route Partnership B requests solely to the current option pack. This configuration is shown in the following diagram:




## Perform an Option Pack Upgrade

Consider the following points before upgrading the Web Agent Option Pack:

- The Web Agent Option Pack version must be compatible with the Policy Server version. To learn which Policy Server versions the Web Agent Option Pack is compatible with, see the CA Single Sign-On Platform Support Matrix.

- When different compatible versions of the Web Agent Option Pack and Policy Server are mixed, federation functionality is limited to that of the lesser of the two components.
- If the Web Agent and Web Agent Option Pack are installed on the same system, they must be the same version, including the Service Pack and CR version.

**Important!** When upgrading, the program automatically creates new back-up configuration files and overwrites the existing configuration files.

**To upgrade the Web Agent Option Pack**

1. Run the Web Agent Option Pack installation program. The installation program can be run as an upgrade.
   For more information, see the installation instructions (see page 887).
2. Follow the prompts to upgrade the system.

# SiteMinder Agent for JBoss

The following sections detail how to install and configure the agent for JBoss.

# Agent for JBoss Introduced

**Contents**



## Introduction

The Agent for JBoss integrates with the JBoss Application Server to secure J2EE resources deployed on that operating environment. The CA Single Sign-on Agent for JBoss provides the following two JBossSX custom security interceptors that allow it to be configured into CA Single Sign-on and CA Single Sign-On Web Services Security environments as required:

- **Agent Security Interceptor**
  The Agent Security Interceptor provides a CA Single Sign-on Agent solution that provides access control for web application resources (including servlets, HTML pages, JSP, and image files).

- **WSS Agent Security Interceptor**
  The WSS Agent Security Interceptor provides a Web Services Security (WSS) Agent solution that provides CA Single Sign-On Web Services Security access control for JAX-WS and JAX-RPC web service resources.

## Required Background Information

This document is not intended for users who are new to Java, J2EE standards, or application server technology and assumes that you have the following technical knowledge:

- An understanding of J2EE application servers and multi-tier architecture.
- Familiarity with Java Authentication and Authorization Server (JAAS) and the JBossSX security framework.
- Knowledge of how to provide security constraints for J2EE components through security realms and deployment descriptors.
- Experience with configuring and managing the JBoss Application Server.
- If configuring protection for web applications, familiarity with CA Single Sign-on concepts and terms.
- If configuring protection for web services, understanding of JAX-RPC and JAX-WS web service implementations and handlers and familiarity with Web Services Security concepts and terms.
- Knowledge of Policy Server configuration tasks.

## Agent Security Interceptor

The CA Single Sign-on Agent Security Interceptor provides an *identity assertion* solution for securing JBoss web container resources by perimeter authentication.

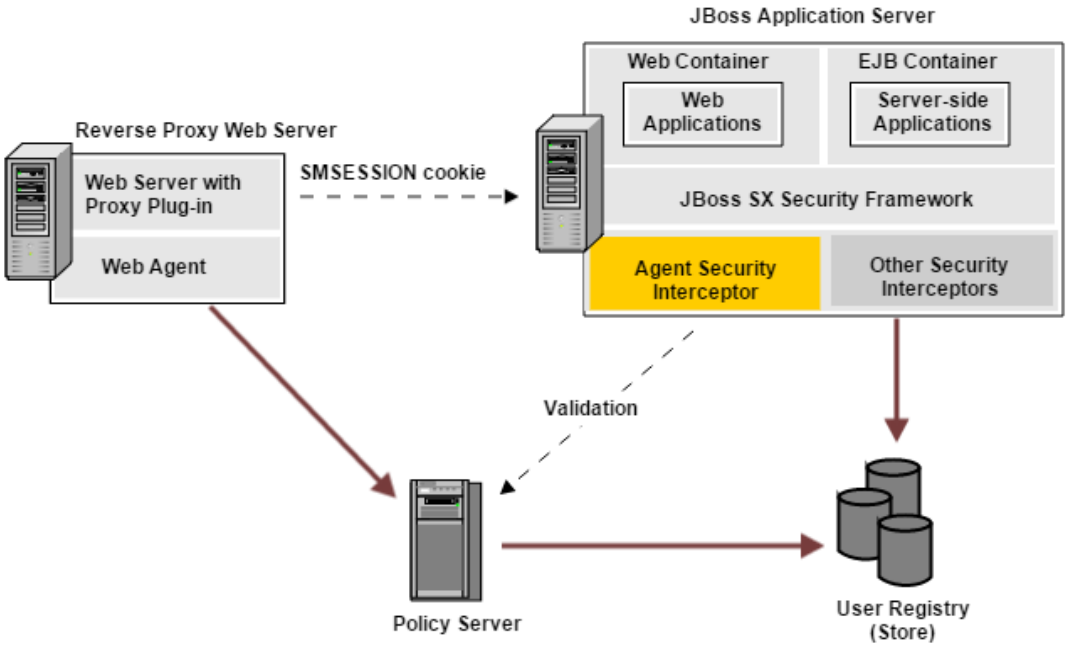
In the perimeter authentication model, user identity is validated outside the JBoss security domain and passed to the JBoss Application Server in the form of a token associated with the user request. An Identity Asserter configured within the JBoss security domain then obtains authenticated user information from the token.

The Agent Security Interceptor allows the JBoss Application Server to trust requests with associated CA Single Sign-on session cookies (SMSESSION) so that these users are not challenged for credentials.

Session cookies are obtained from a Web Agent on a proxy server configured to:

- Intercept HTTP requests for JBoss resources
- Authenticate and authorize users through policies defined on the Policy Server
- Forward requests together with user credentials (in a session cookie) to the application server.

The following illustration shows the Web Agent and proxy server:




When you configure the Agent Security Interceptor as an identity asserter in a security realm, the JBossSX security framework passes any session cookies associated with a request for a resource within that realm to the Agent Security Interceptor for validation. The Agent Security Interceptor then:

1. Validates the token by calling the Policy Server to verify that its session is valid ( session cookie).
2. Obtains the requester userDN from the token and maps it to a username.
3. Passes the associated username and session information back to the JBossSX security framework.

**Note:** If you must only allow access to web applications for clients with *existing* Single Sign-On (SSO) sessions, use the Agent Security Interceptor as a standalone component without the proxy server-related components.

## Agent Security Interceptor Components

The CA Single Sign-on Agent Security Interceptor consists of the following modules that you can configure into the JBossSX security framework:

### Agent Authenticators

In the JBossSX security framework, requests for web application resources in the web container are handled by default authenticators for Basic, Client-Cert, Form, and Digest authentication.

The Agent Security Interceptor provides the following custom replacement Agent Authenticators that extend the functionality of the JBoss default authenticators with the ability to authenticate a user request based on an associated session cookie:

- **SMJBossIdentityAsserter**
  (New) Authenticates user identity using the session cookie only. If there is no valid session cookie, the authenticator returns an authentication failure result.

- **SMJBossBasicAuthenticator**
  (Replaces JBoss default BasicAuthenticator) First attempts to authenticate user identity using the session cookie. If there is no valid session cookie, performs Basic authentication.

- **SMJBossFormAuthenticator**
  (Replaces JBoss default FormAuthenticator) First attempts to authenticate user identity using the session cookie. If there is no valid session cookie, performs Form authentication.

- **SMJBossClientCertAuthenticator**
  (Replaces JBoss default ClientCertAuthenticator) Authenticates the user identity using the session cookie. A session cookie is required.

- **SMJBossDigestAuthenticator**
  (Replaces JBoss default DigestAuthenticator) Authenticates the user identity using the session cookie. A session cookie is required.

**Note:** The SMJBossClientCertAuthenticator and the SMJBossDigestAuthenticator require a session (SMSESSION cookie) to authenticate users.

The Agent Authenticators first attempt to retrieve a session cookie from a request. If there is a valid session cookie, the Agent Login Module is used to authenticate the user and create user principles. If there is no valid session cookie, the appropriate JBossSX default authenticator functionality occurs.

### Agent Login Module

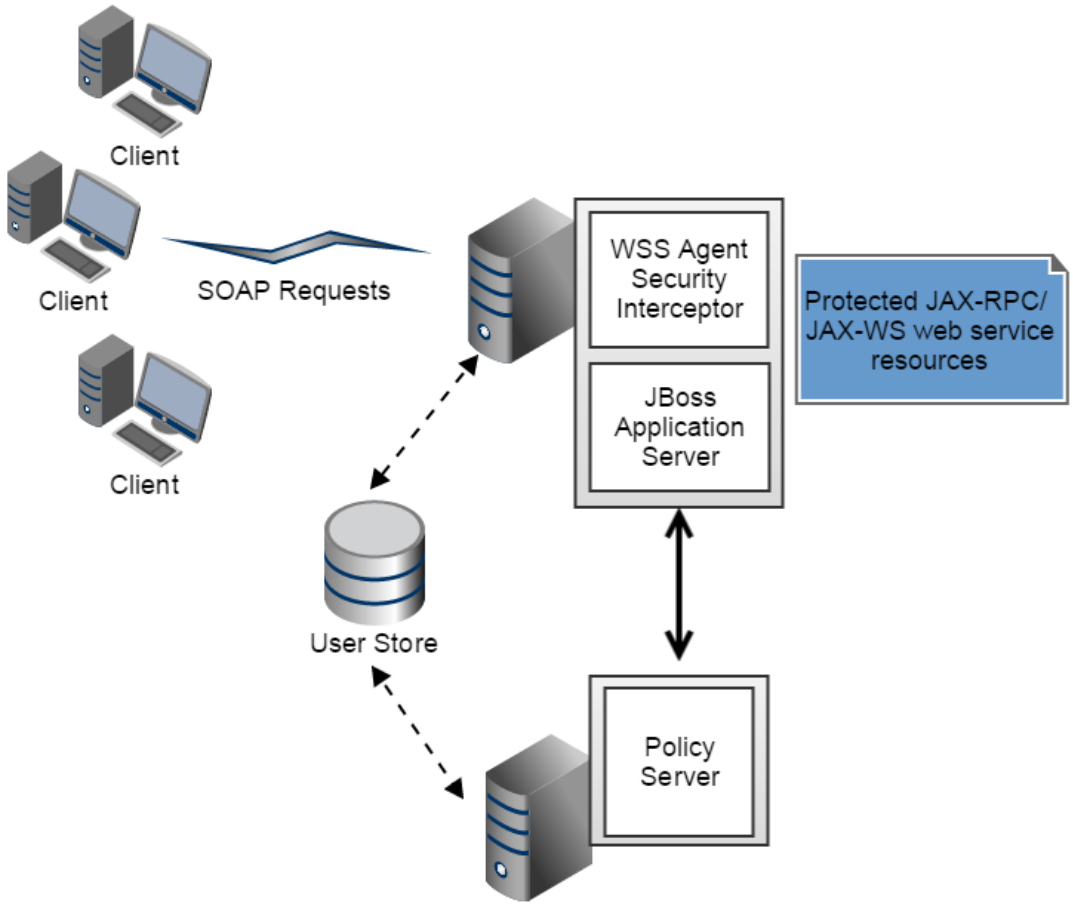
The CA Single Sign-on Agent Login Module authenticates credentials (username/password) obtained from valid session cookies by CA Single Sign-on Agent authenticators.

If authentication is successful, the Agent Login Module populates a JAAS Subject with a CA Single Sign-on Principal that contains the username and associated session data.

## WSS Agent Security Interceptor

The WSS Agent Security Interceptor provides a WSS Agent solution for the JBoss Application Server. The WSS Agent Security Interceptor integrates the JBoss Application Server into the CA Single Sign-On Web Services Security environment, enabling you to implement policy-based fine-grained access control to protect JBoss-hosted JAX-RPC and JAX-WS web service resources. The WSS Agent Security Interceptor also supports bi-directional CA Single Sign-On Web Services Security/CA Single Sign-on and JBoss single sign-on (SSO).

A high-level overview of the WSS Agent Security Interceptor architecture is shown in the following illustration:




When fully configured into the JBossSX security infrastructure, the WSS Agent Security Interceptor does the following:

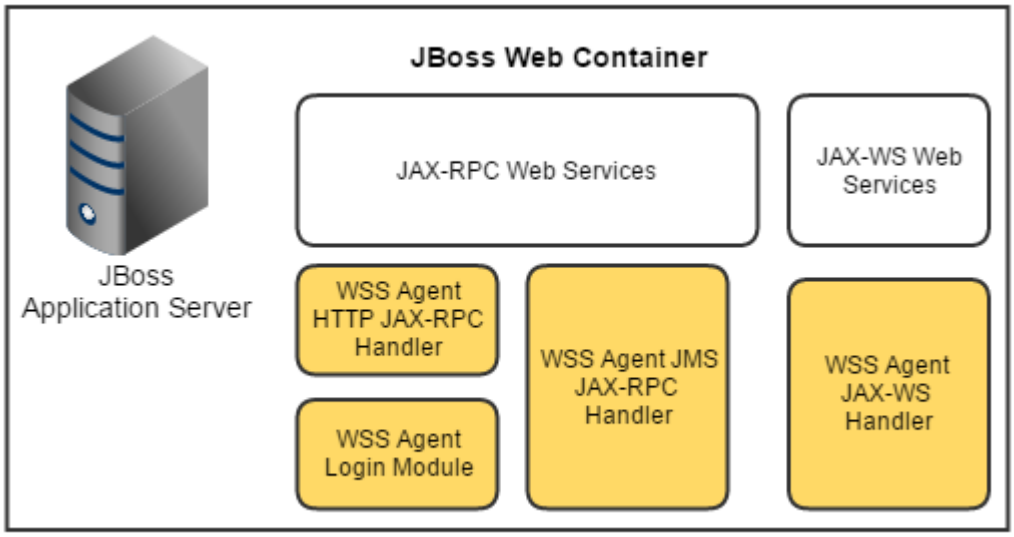
1. Intercepts SOAP requests sent over HTTP(S) or JMS transports to JAX-RPC and JAX-WS web services deployed on the JBoss Application Server.
2. Communicates with the Policy Server to authenticate and authorize the message sender
3. Upon successful authentication and authorization, passes the request message on to the addressed web service.

## WSS Agent Security Interceptor Components

The WSS Agent Security Interceptor consists of the following modules that you can configure into the JBossSX security framework:

- WSS Agent JAX-WS Handler
- WSS Agent JMS JAX-RPC Handler
- WSS Agent HTTP JAX-RPC Handler
- WSS Agent Login Module

**Note:** You do not need to configure all WSS Agent modules, only the ones you require. WSS Agent modules can be configured globally for all web services of each type or for each individual web service.




## WSS Agent JAX-WS Handler

The WSS Agent JAX-WS Handler is a custom JAX-WS Handler that intercepts requests for JAX-WS web services and authenticates credentials obtained from intercepted requests against associated user directories configured in CA Single Sign-On Web Services Security:

**Note:** The WSS Agent JAX-WS Handler can obtain credentials from SOAP requests or from associated session cookies of users with pre-established CA Single Sign-On Web Services Security and sessions.

If Web Services Security authentication is successful, the WSS Agent JAX-WS Handler determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated Web Services Security authorization policies.

## WSS Agent JMS JAX-RPC Handler

The WSS Agent JMS JAX-RPC Handler is a custom JAX-RPC Handler that intercepts requests for JAX-RPC web services sent over JMS transport and authenticates credentials obtained from those requests against user directories configured in Web Services Security.

If Web Services Security authentication is successful, the WSS Agent JMS JAX-RPC Handler determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated Web Services Security authorization policies.

### WSS Agent HTTP JAX-RPC Handler

The WSS Agent HTTP JAX-RPC Handler is a custom JAX-RPC Handler that intercepts SOAP message requests sent to JAX-RPC web services over HTTP transport and diverts them to the WSS Agent Login Module for authentication and authorization decisions.

**Note:** If you configure the WSS Agent JAX-RPC Handler, you must also configure the WSS Agent Login Module.

### WSS Agent Login Module

The WSS Agent Login Module is a JAAS Login Module that performs authentication and authorization for JAX-RPC web services protected by the WSS Agent HTTP JAX-RPC Handler. (Login Module functionality is built into the WSS Agent WS and JMS JAX-RPC Handlers.)

The WSS Agent Login Module can authenticate and authorize credentials obtained by the WSS Agent JAX-RPC Handler from SOAP requests or from associated session cookies of user with pre-established Web Services Security and sessions.

If Web Services Security authentication is successful, the WSS Agent Login Module determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated Web Services Security authorization policies.

**Note:** If you configure the WSS Agent Login Module, you must also configure the WSS Agent JAX-RPC Handler.

# Install the Agent for JBoss

This section contains the following topics:



## Install Preparation

**Contents**

Before you install a CA Single Sign-on Agent for JBoss, there are a number of pieces of information you will need and requirements that must be met.

## Locate the Platform Support Matrix

Use the Platform Support Matrix to verify that the operating environment and other required third-party components are supported.

**Follow these steps:**

1. Log in to the CA Support site (http://www.ca.com/support).
2. Locate the Technical Support section.
3. Enter CA Single Sign-On in the Product Finder field.
   The CA Single Sign-On product page appears.
4. Click Product Status, CA Single Sign-On Family of Products Platform Support Matrices.
5. Locate the **CA Single Sign-on Agent for Application Servers 12.52** entry and open open the associated PDF file.

**Note:** You can download the latest JDK and JRE versions at the Oracle Developer Network ( http://www.oracle.com/technetwork/java/index.html).

## Software Requirements

**General Requirements**

Supported versions of the following software are always required before you install the CA Single Sign-on Agent.

- JBoss Enterprise Application Platform. For hardware and software requirements, see the JBoss Enterprise Application Platform documentation.
- *One* of the following Policy Servers:
  - CA Single Sign-on Policy Server (for web application protection)
  - Policy Server (for web service and, if also licensed for CA Single Sign-on, web application protection)

- Java virtual machine (JVM) with the path to the JVM present in the host environment. For example, on UNIX systems, if the JVM is not in the PATH variable, run the following commands:

```
PATH=$PATH:JVM/bin
export PATH
```

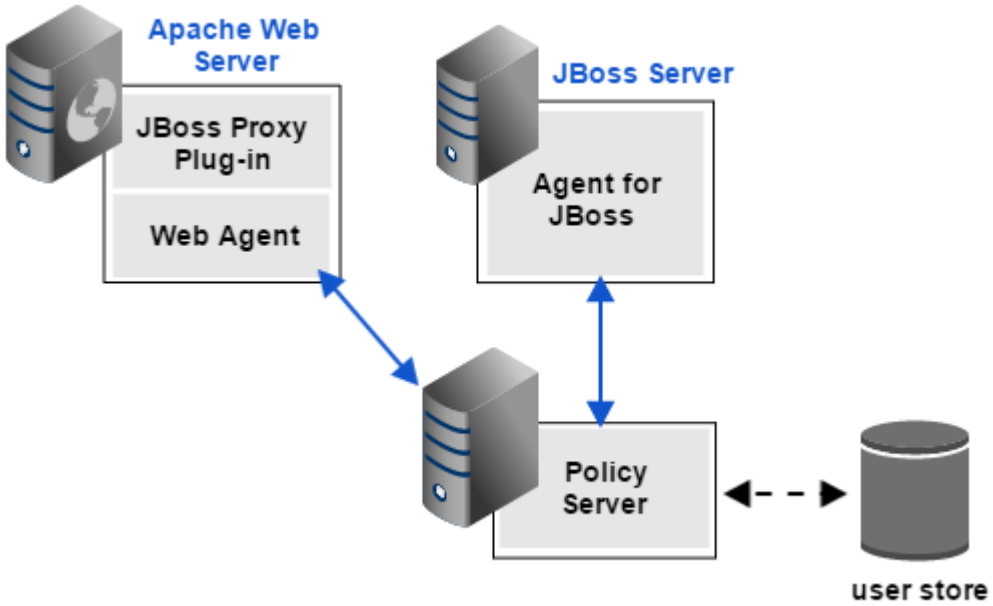
  - ***JVM***
    Specifies the location of your Java virtual machine (for example /opt/jre1.5.0_06/bin).

**Additional Requirements for the CA Single Sign-on Agent Web Interceptor**

To use the CA Single Sign-on Agent Web Interceptor to validate identities obtained from CA Single Sign-on session cookies during perimeter authentication, the following software is also required:

- CA Single Sign-on Web Agent
- A web server and proxy plug-in supported by CA Single Sign-on and JBoss
  For supported web servers and proxy plug-ins, see:
  - Platform Support Matrices on the Technical Support (http://www.ca.com/support) site.
  - Supported Configurations for JBoss Enterprise Application Platform in the JBoss Enterprise Application Platform documentation.

The following illustration shows where each of these software components is installed in an environment that uses CA Single Sign-on SSO-based perimeter authentication:




For a complete list of supported software, operating systems, Java environments, and prerequisite A product versions, refer to the CA Single Sign-on Agent for Application Servers Platform Support Matrix on the Technical Support site (http://www.ca.com/support).

## Installation Location References

The following references to the installed location of CA Single Sign-on Agent and JBoss software are used throughout this document.

- ***SMAGENT_HOME***
  Refers to the installed location of the CA Single Sign-on Agent for JBoss.
  The default location is:
  - C:\Program Files\CA\JBossAgent (Windows)
  - /CA/JBossAgent (UNIX)
- ***JBOSS_HOME***
  Refers to the installed location of the JBoss Application Server.
  For example, the default location for JBoss Enterprise Application Platform 4.3 is:
  - C:\jboss-eap-4.3\jboss-as on Windows
  - /jboss-eap-4.3/jboss-as on UNIX

## Information Required During CA Single Sign-on Agent Installation

The CA Single Sign-on Agent for JBoss installation program prompts you to supply the following information:

- Location of the JVM to use.
- Location of the JBoss Application Server installation. For example, the default for JBoss Enterprise Application Platform 4.3 is C:\jboss-eap-4.3\jboss-as on Windows and /jboss-eap-4.3/jboss-as on UNIX.
- If you proceed to configure the Agent, the configuration wizard prompts you for the following additional information:
  - Policy Server IP Address
  - Information about the Trusted Host:
    To register a new Trusted Host, you need the name of the Trusted Host Configuration Object that you created when you configured the CA Single Sign-on Policy Server for the CA Single Sign-on agent providers.

    **Note:** If you want to register a new Trusted Host, be sure that the Policy Server is running before you start the CA Single Sign-on Agent installation.

    To use an existing Trusted Host on the physical computer where the CA Single Sign-on Agent resides, you need the location of the SmHost.conf file.
  - Agent Configuration Object name for the Agent you created when you configured the CA Single Sign-on Policy Server for the CA Single Sign-on agent providers

## Preconfigure Policy Objects for the CA Single Sign-on Agent

This section describes how to preconfigure policy objects for the CA Single Sign-on Agent for JBoss on the Policy Server.

## Policy Object Preconfiguration Overview

Before you install the CA Single Sign-on Agent for JBoss, the Policy Server must be installed and be able to communicate with the system where you plan to install the CA Single Sign-on Agent. Additionally, configure the Policy Server with the following:

- **An administrator that has the right to register trusted hosts**
  A trusted host is a client computer where one or more CA Single Sign-on Agents are installed. The term trusted host refers to the physical system. There must be an administrator with permission to register trusted hosts with the Policy Server.

- **Agent object/Agent identity**
  An Agent object creates an Agent identity by assigning the Agent a name. You define an Agent identity from the Agents object in the Administrative UI. You assign the Agent identity a name and specify the Agent type as a Web Agent.
  The name you assign for the Agent is the same name you specify in the DefaultAgentName parameter for the Agent Configuration Object that you must also define to centrally manage an Agent.

- **Host Configuration Object**
  This object defines the communication between the trusted host and the Policy Server after the initial connection between the two is made.
  A trusted host is a client computer where one or more CA Single Sign-on Agents can be installed. The term trusted host refers to the physical system, in this case the JBoss Application Server host. Do not confuse this object with the trusted host's configuration file, SmHost.conf, which is installed at the trusted host after a successful host registration. The settings in the SmHost.conf file enable the host to connect to a Policy Server for the first connection only. Subsequent connections are governed by the Host Configuration Object.

- **Agent Configuration Object**
  This object includes the parameters that define the CA Single Sign-on Agent configuration. There are a few required parameters you must set for basic operation.
  The Agent Configuration Object must include a value for the DefaultAgentName parameter. This entry should match an entry you defined in the Agent object.

## Preconfigure the Policy Objects

The following is an overview of the configuration procedures to perform on the Policy Server before installing the Agent software:

1. Duplicate or create a new Host Configuration Object, which holds initialization parameters for a Trusted Host. (If upgrading from an earlier Agent install, you can use the existing Host Configuration object).
   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

2. As necessary, add or edit Trusted Host parameters in the Host Configuration Object that you just created.

3. Create an Agent identity for the CA Single Sign-on Agent for JBoss. Select **Web Agent** as the Agent type for the CA Single Sign-on Agent for JBoss.

   **Note:** If you are using CA Single Sign-on SSO-based perimeter authentication to validate identities obtained from CA Single Sign-on session cookies, configure separate Agents identities for the CA Single Sign-on Agent for JBoss and the Web Agent on the proxy server.

4. Duplicate an existing or create a new Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

5. In the Agent Configuration Object you created, verify that the DefaultAgentName parameter is set to specify the Agent identity defined in Step 3.

### Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files are in the following locations:

- Windows
  *jre_home*\lib\security
- UNIX
  *jre_home*/lib/security

*jre_home d*efines the location of your Java Runtime Environment installation.

## Install a SiteMinder Agent on a Windows System

**Contents**



The following sections describe how to install the CA Single Sign-on Agent on a Windows system.

### Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

**Follow these steps:**

1. Open the Windows Control Panel.
2. Double-click System.
3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

## Run the Installation on Windows

Install the CA Single Sign-on Agent for JBoss using the using the installation media on the Technical Support site.

**Note**: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

**Follow these steps:**

1. Exit all applications that are running.
2. Navigate to where the installation executable is located.
3. Double-click ca-sm-jboss-version-*cr*-win32.exe.

   - ***cr***
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

   The CA Single Sign-on Agent for JBoss installation wizard starts.
4. Use gathered system and component information to install the CA Single Sign-on Agent. Consider the following when running the installer:

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
5. Review the information on the Pre-Installation Summary page, then click Install.

   **Note:** The installation program may detect that newer versions of certain system dlls are installed on your system. It asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

The CA Single Sign-on Agent files are copied to the specified location. Afterward, the CA CA Single Sign-on Agent for JBoss Configuration screen is displayed.

6. Choose one of the following options:
   - Yes. I would like to configure the CA CA Single Sign-on Agent for JBoss now.
   - No. I will configure the CA CA Single Sign-on Agent for JBoss later.
7. Click Done.
   If you selected the option to configure the Agent now, the installation program prepares the CA CA Single Sign-on Agent for JBoss Configuration Wizard and begins the trusted host registration and configuration processes.
   Do the following:
   - Register the trusted host. You can do this before or after configuring an Agent, but the Agent will *not* be able to communicate properly with the Policy Server unless the trusted host is registered.
   - Configure the CA Single Sign-on Agent.

   If you did not select the option to configure the Agent now, the installation program prompts you to restart your system.Select whether to restart the system automatically or later on your own.

**Installation Notes:**

- After installation, you can review the installation log file in *SMAGENT_HOME*\install_config_info. The file name is: CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log
  - ***SMAGENT_HOME***
    Specifies the path to where the CA Single Sign-on Agent is installed.
    **Default:** C:\Program Files\CA\JBossAgent
- You may choose not to start the CA Single Sign-on Agent for JBoss Configuration Wizard immediately after installation or you may have to reboot your machine after installation. If so, you can start the Wizard manually when you are ready to configure an Agent.

## Install the CA Single Sign-on Agent Using the Unattended Installer on Windows

Once the CA Single Sign-on Agent is installed on one system, you can reinstall it on the same system or install it with the same options on another system using an unattended installation mode. An unattended installation lets you install or uninstall the agent without any user interaction

The unattended installation uses the ca-jboss-agent-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, and so on.

The ca-jboss-agent-installer.properties is located in *SMAGENT_HOME*\install_config_info.

**Follow these steps:**

1. From a system where the agent is already installed, copy the ca-jboss-agent-installer.properties file to a local directory on your system.

2. Download the agent installation media from the Technical Support site.

   **Note**: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

3. Copy the installation media into the same local directory as the ca-jboss-agent-installer.properties file.

4. Open a console window and navigate to the location where you copied the files.

5. Run the following command:

   ```
   ca-sm-jboss-version-cr-win32.exe  -f ca-jboss-agent-installer.properties -
   i silent
   ```

   - ***cr***
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

   The -i silent setting instructs the installer to run in the unattended installation mode. When running this command, if the ca-jboss-agent-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces. For example:

   ```
   ca-sm-jboss-version-cr-win32.exe -f "C:
   \Program Files\CA\JBossAgent\install_config_info\ca-jboss-agent-installer.
   properties " -i silent
   ```

   An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-on Agent installer has begun. The installer uses the parameters specified in the ca-jboss-agent-installer.properties file.

**Note:** To stop the installation manually, open the Windows Task Manager and stop the *installation_media* process.

To verify that the unattended installation completed successfully, see the CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*\install_config_info directory. This log file contains the results of the installation.

## Install a SiteMinder Agent on a UNIX System

**Contents**

- Configure the JVM to Use the JSafeJCE Security Provider (see page 943)

The following sections describe how to install the CA Single Sign-on Agent on a UNIX system.

## Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

**To set the JRE in the PATH variable**

1. Open a Command Window.
2. Run the following commands:

   ```
   PATH=$PATH:JRE
   export PATH
   ```

   - ***JRE***
     Defines the location of your Java Runtime Environment bin directory.

## Run the Installer in GUI Mode on UNIX

Install the CA Single Sign-on Agent for JBoss using the installation media on the Technical Support site.

**Note**: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

**Follow these steps:**

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located
3. If necessary, add executable permissions to the install file by running the following command:

   ```
   chmod +x installation_media
   ```

   - ***installation_media***
     Specifies the CA Single Sign-on Agent installer executable
4. Enter the following command:

   ```
   sh ./ca-sm-jboss-version-cr-unix_version.bin
   ```

   - ***cr***
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.
   - ***unix_version***
     Specifies the UNIX version: **sol** or **linux**.

The CA Single Sign-on Agent for JBoss installation wizard starts.

5. Use gathered system and component information to install the CA Single Sign-on Agent. Consider the following when running the installer:

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - Do not use space characters in the CA Single Sign-on WSS Agent install path. For example, " /CA Technologies/agent" will result in install failure.

6. Review the information displayed on the Pre-Installation Summary page, then click Install.

   **Note:** If the installer detects newer versions of certain system libraries installed on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The CA Single Sign-on Agent files are copied to the specified location. Afterward, the CA CA Single Sign-on Agent for JBoss Configuration screen is displayed.

7. Choose one of the following options:

   - Yes. I would like to configure the CA CA Single Sign-on Agent for JBoss now.
   - No. I will configure the CA CA Single Sign-on Agent for JBoss later.

8. Click Done.
   If you selected the option to configure the Agent now, the installer prepares the CA CA Single Sign-on Agent for JBoss Configuration Wizard and begins the host registration and configuration processes.
   Do the following:

   - Register the trusted host. You can perform this process before or after configuring an Agent. However the Agent *cannot* communicate properly with the Policy Server unless the trusted host is registered.
   - Configure the CA Single Sign-on Agent.

   If you did not select the option to configure the Agent now, the installation program prompts you to restart your system. Select whether to restart the system automatically or later on your own.

**Installation Notes:**

- After installation, you can review the installation log file in *SMAGENT_HOME/*install_config_info. The file name is: CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log

- ***SMAGENT_HOME***
  Specifies the path to where the CA Single Sign-on Agent is installed.

- If you do not start the configuration wizard immediately after installation, you can start the Wizard manually when you are ready to configure an Agent.

- If you must reboot the server after installation, you can start the Wizard manually when you are ready to configure an Agent.

## Run the Installer in Console Mode on UNIX

Install the CA Single Sign-on Agent for JBoss using the installation media on the Technical Support site.

**Note**: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

**Follow these steps:**

1. Exit all applications that are running.

2. Open a shell and navigate to where the install program is located

3. If necessary, add executable permissions to the install file by running the following command:

   ```
   chmod +x installation_media
   ```

   - ***installation_media***
     Specifies the CA Single Sign-on Agent installer executable

4. Enter the following command:

   ```
   sh ./ca-sm-jboss-version-cr-unix_version.bin -i console
   ```

   - ***cr***
     Specifies the cumulative release number. The base release does not include a cumulative release number in the file name.

   - ***unix_version***
     Specifies the UNIX version: **sol** or **linux**.

   The CA Single Sign-on Agent for JBoss installation wizard starts.

5. Use gathered system and component information to install the CA Single Sign-on Agent. Consider the following as you make your selections:

   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

- Do not use space characters in the CA Single Sign-on WSS Agent install path. For example, " /CA Technologies/agent" will result in install failure.

6. Review the information displayed on the Pre-Installation Summary page, then proceed.

**Note:** If the installer detects newer versions of certain system libraries installed on your system, it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

The CA Single Sign-on Agent files are copied to the specified location. Afterward, the CA CA Single Sign-on Agent for JBoss Configuration page is displayed.

7. Select whether to restart the system now or later on your own.

8. Hit Enter.

**Note:** After installation, you can review the installation log file in *SMAGENT_HOME/* install_config_info. The file name is: CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog. log.

## Install the CA Single Sign-on Agent Using the Unattended Installer on UNIX

Once the CA Single Sign-on Agent is installed on one system, you can reinstall it on the same system or install it with the same options on another system using an unattended installation mode. An unattended installation lets you install or uninstall the agent without any user interaction

The unattended installation uses the ca-jboss-agent-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, and so on.The ca-jboss-agent-installer.properties is located in *SMAGENT_HOME/* install_config_info.

**Follow these steps:**

1. From a system where the CA Single Sign-on Agent is already installed, copy the ca-jboss-agent-installer.properties file to a local directory on your system.

2. Download the agent installation media from the Technical Support site.

**Note**: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

1. Copy the installation media into the same local directory as the ca-jboss-agent-installer. properties file.

2. Open a console window and navigate to the location where you copied the files.

3. Run the following command:

   ```
   ca-sm-jboss-version-cr-unix_version.bin -f ca-jboss-agent-installer.properties -
   i silent
   ```

   - ***cr***
     Specifies the cumulative release number. The base version does not include a cumulative release number in the file name.

   - ***unix_version***
     Specifies the UNIX version: **sol** or **linux**.

   The -i silent setting instructs the installer to run in the unattended installation mode.
   When running this command, if the ca-jboss-agent-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.
   For example:
   ca-sm-jboss-version-*cr-unix_version*.bin -f "/CA/JBossAgent/install_config_info/ca-jboss-agent-installer.properties " -i silent
   The -i silent setting instructs the installer to run in the unattended installation mode.
   An InstallAnywhere status bar appears, which shows that the unattended CA Single Sign-on Agent installer has begun. The installer uses the parameters specified in the ca-jboss-agent-installer.properties file.

**Note:** To stop the installation manually, type Ctrl+C.

To verify that the unattended installation completed successfully, see the CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME/*install_config_info directory. This log file contains the results of the installation.

## Configure the JVM to Use the JSafeJCE Security Provider

The WSS Agent XML encryption function requires that you configure the JVM to use the JSafeJCE security provider.

**Follow these steps:**

1. Navigate to the java.security file and open the file for editing. The java.security file is in the following location:

   - *JVM_HOME*\jre\lib\security (Windows)

   - *JVM_HOME*/jre/lib/security (UNIX)

     *JVM_HOME* is the installed location of the JVM used by the application server.

2. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE). Place the JSafeJCE security provider immediately after the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE).

```
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

The initial FIPS mode does not affect the final FIPS mode you select for the WSS Agent.

3. Add the following line to set the *initial* FIPS mode of the JsafeJCE security provider. Place this line anywhere in the file.

The following example shows the addition of the two entries for the JsafeJCE security provider.

```
security.provider.1=com.ibm.crypto.provider.IBMJCE
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=com.ibm.security.sasl.IBMSASL
security.provider.11=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.12=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.13=org.apache.harmony.security.provider.PolicyProvider
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

## How to Configure the Agent and Register A System as a Trusted Host on Windows

**Contents**



A *trusted host* is a client computer where one or more CA Single Sign-on or SOA Agents can be installed. The term trusted host refers to the physical system.

To establish a connection between the trusted host and the Policy Server, you need to register the host with the Policy Server. After registration is complete, the registration tool creates the SmHost.conf file. After this file is created successfully, the client computer becomes a trusted host.

### Gather Information Required for CA Single Sign-on WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

> **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  > **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Configure Agents and Register Your System as a Trusted Host

You can configure your CA Single Sign-on Agent and register a trusted host immediately after installing the agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a CA Single Sign-on Agent on your system.

**To configure Agents and register a trusted host**

1. If necessary, start the CA Single Sign-on Configuration Wizard. The default method is to select Start, Programs, CA, CA Single Sign-on, CA Single Sign-on Configuration Wizard. If you have placed the Wizard shortcut in a non-default location, the procedure will be different. (Alternatively, navigate to *SMAGENT_HOME*\install_config_info and run ca-jbossagent-config.exe.)

   **Note:** If you chose to configure the CA Single Sign-on Agent immediately after the installation, the installer automatically starts the Configuration Wizard.The CA Single Sign-on Configuration Wizard starts.

2. Use gathered system and component information to configure the CA Single Sign-on Agent and register the host.

   **Note:** If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *SMAGENT_HOME*\config. You can modify this file.

## Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of the CA Single Sign-on Agent for JBoss, check the CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*/install_config_info directory.

## Modify the SmHost.conf File

CA Single Sign-on Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *SMAGENT_HOME*\config directory.

2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

> **Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

- **hostconfigobject**

  Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

  If you want to change the host configuration object an object so the CA Single Sign-on Agent uses it, you need to modify this setting.

  **Example:** hostconfigobject="*host_configuration_object*"

- **policyserver**

  Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

  "*IP_address*, *port*,*port*,*port*"

  The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

  To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

  Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

  > **Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

  **Default:***IP_address*, 44441,44442,44443

**Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

**Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

  Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

  **Default:** 60

  **Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool

When you install an agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent Agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *SMAGENT_HOME*\bin directory when you install the CA Single Sign-on Agent.

**Follow these steps:**

1. Open a command prompt window.
2. Enter the smreghost command using the following required arguments:

```
smreghost -i policy_server_IP_address:[port]
-u administrator_username -p Administrator_password
-hn hostname_for_registration -hc host_configuration_ object
```

**Note:** If the "-p Administrator_password" argument is not specified in the smreghost command, you are prompted to specify the password.

**Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

- **-i *policy_server_IP_ address:port***
  Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.
  If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.
  Use a colon between the IP address and non-default port number, as shown in the following examples.
  **Default:** (ports) 44441,44442,44443
  **Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555
  **Example**: (IPv4 default ports) -i 127.0.0.1
  **Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]
  **Example:** (IPv6 default ports) -i [2001:DB8::/32]

- **-u *administrator_username***

  Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

  Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**
  Specifies whether the shared secret is updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

  COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

  If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

  **Default:** COMPAT

  **Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

### Register Multiple Trusted Hosts on One System

You typically register only one trusted host for each machine where application servers and CA Single Sign-on or CA Single Sign-on WSS Agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by CA Single Sign-on or CA Single Sign-on WSS Agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  **Note:** If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads:

  "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

## How to Configure the Agent and Register a System as a Trusted Host on UNIX

**Contents**

A *trusted host* is a client computer where one or more CA Single Sign-on or SOA Agents can be installed. The term trusted host refers to the physical system.

To establish a connection between the trusted host and the Policy Server, you need to register the host with the Policy Server. After registration is complete, the registration tool creates the SmHost. conf file. After this file is created successfully, the client computer becomes a trusted host.

## Gather Information Required for CA Single Sign-on WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

- **SM Admin User Name**
  The name of a Policy Server administrator allowed to register the host with the Policy Server. This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator user name is siteminder .

- **SM Admin Password**
  The Policy Server administrator account password.

- **Trusted Host Name**
  Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

  **Note:** This name must be unique among trusted hosts and not match the name of any other Agent.

- **Host Configuration Object**
  The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

  **Note:** This value must match the Host Configuration Object entry preconfigured on the Policy Server.

- **Policy Server IP Address**
  The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used. You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:
  Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)
  Note also that if you specify a non-default port, that port is used for the Policy Server's authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:
  policyserver="*ip_address*,5555,5555,5555"

- **FIPS Encryption Mode**
  Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

  - **FIPS Compatibility Mode (Default)**
    Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - **FIPS Only Mode**
    Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.
    **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

## Configure Agents and Register a Trusted Host in GUI or Console Mode

You can configure CA Single Sign-on Agents and register a trusted host immediately after installing the CA Single Sign-on Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

**Note:** You only register the host once, *not* each time you install and configure a CA Single Sign-on Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.
- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

**To configure Agents and register a trusted host**

1. If necessary, start the Configuration Wizard as follows:
   a. Open a console window.
   b. Navigate to *SMAGENT_HOME*/install_config_info, where *agent_home* is the installed location of the CA Single Sign-on Agent.

c. Enter one of the following commands:
GUI Mode: ./ca-jbossagent-config.bin
Console Mode: ./ca-jbossagent-config.bin -i console

The Configuration Wizard starts.

2. Use gathered system and component information to configure the CA Single Sign-on Agent and register the host.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *SMAGENT_HOME*/config. You can modify this file.

## Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of the CA Single Sign-on Agent for JBoss, check the CA_CA Single Sign-on®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*/install_config_info directory.

## Modify the SmHost.conf File

CA Single Sign-on Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

**To modify the SmHost.conf file**

1. Navigate to the *SMAGENT_HOME/*config directory.

2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

**Important!** Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

- **hostconfigobject**

  Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

  If you want to change the host configuration object an object so the CA Single Sign-on Agent uses it, you need to modify this setting.

  **Example:** hostconfigobject="*host_configuration_object*"

- **policyserver**

  Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

  "*IP_address*, *port*,*port*,*port*"

  The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

  To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

  Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA Single Sign-On environment or is no longer in service, delete the entry.

  **Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

  **Default:***IP_address*, 44441,44442,44443

  **Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

  **Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:** policyserver="123.122.1.1, 44441,44442,44443"policyserver="111.222.2.2, 44441,44442,44443"policyserver="321.123.1.1, 44441,44442,44443"

- **requesttimeout**

  Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

  **Default:** 60

  **Example:** requesttimeout="60"

4. Save and close the SmHost.Conf file.
   The changes to the SmHost.conf file are applied.

## Re-register a Trusted Host Using the Registration Tool

When you install a CA Single Sign-on Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent Agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA Single Sign-On environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *SMAGENT_HOME/*bin directory when you install the CA Single Sign-on Agent.

**To re-register a trusted host using the registration tool**

1. Open a command prompt window.
2. Ensure that the library path environment variable contains the path to the CA Single Sign-on Agent's bin directory by entering the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH:agent_home/bin
   export LD_LIBRARY_PATH
   ```

   For example:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/CA/JBossAgent/bin
   export LD_LIBRARY_PATH
   ```

3. Enter the smreghost command using the following required arguments:

   ```
   smreghost -i policy_server_IP_address:[port]
   -u administrator_username -p Administrator_password
   -hn hostname_for_registration -hc host_configuration_ object
   ```

   **Note:** If the "-p Administrator_password" argument is not specified in the smreghost command, you are prompted to specify the password.

   **Note:** Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

- **-i *policy_server_IP_ address:port***
  Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.
  If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.
  Use a colon between the IP address and non-default port number, as shown in the following examples.
  **Default:** (ports) 44441,44442,44443
  **Example:** (IPv4 non-default port of 55555) -i 127.0.0.1:55555
  **Example**: (IPv4 default ports) -i 127.0.0.1
  **Example:** (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]
  **Example:** (IPv6 default ports) -i [2001:DB8::/32]

- **-u *administrator_username***

  Indicates the name of the CA Single Sign-On administrator with the rights to register a trusted host.

- **-p *Administrator_password***

  Indicates the password of the Administrator who is allowed to register a trusted host.

- **-hn *hostname_for_registration***

  Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

- **-hc *host_config_object***

  Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server beforeyou can register a trusted host.

- **-sh *shared_secret***

  Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

- **-rs**

Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

- **-f *path_to_host_config_file***

  (Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

  If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

- **-cf *FIPS mode***

  Specifies one of the following FIPS modes:

  COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA Single Sign-On encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

  - ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

  **Important!** A CA Single Sign-On installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA Single Sign-On, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

  If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

  **Default:** COMPAT

  **Note:** More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA Single Sign-On Cryptographic Boundary exists in the Policy Server Administration Guide.

- **-o**
  Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

## Register Multiple Trusted Hosts on One System

You typically register only one trusted host for each machine where application servers and CA Single Sign-on or CA Single Sign-on WSS Agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by CA Single Sign-on or CA Single Sign-on WSS Agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  **Note:** If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads:

  "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

## Software Installation for Perimeter Authentication for Agent Security Interceptor

To support perimeter authentication for the CA Single Sign-on Agent Security Interceptor, install and configure the following additional software:

1. Install a supported web server on the proxy server system.
2. Install and configure a supported proxy module on the proxy web server. For detailed proxy module installation and configuration directions, see the JBoss Enterprise Application Platform documentation.
3. Install and configure a Web Agent on the proxy server.

## Uninstall a SiteMinder Agent for JBoss

To uninstall a CA Single Sign-on Agent, run the CA Single Sign-on uninstall wizard.

**To uninstall the CA Single Sign-on Agent on Windows or UNIX systems**

1. Navigate to the *SMAGENT_HOME*\install_config_info (Windows) or *SMAGENT_HOME* /install_config_info (UNIX) directory and run the CA Single Sign-on uninstall wizard:
   - Windows: jbossagent-uninstall.cmd
   - UNIX: jbossagent-uninstall.sh

   The uninstall wizard starts.
2. Confirm that you want to remove the CA Single Sign-on Agent.

The uninstall wizard removes the CA Single Sign-on Agent.

**Note:** You may also want to revert any JBoss configuration files that you modified for the CA Single Sign-on Agent to their previous state.

# Agent for JBoss Configuration Settings

**Contents**



## CA Single Sign-on Agent for JBoss Configuration File

By default, the CA Single Sign-on Agent for JBoss installation creates a single agent configuration file, JavaAgent.conf in the *SMAGENT_HOME/*config directory.

Each Agent configuration file is created with the following required default configuration parameters /values:

| Parameter | Description |
|---|---|
| DefaultAgentName | The agent identity the Policy Server uses to associate policies with the CA Single Sign-on Agent. |
| EnableAgent | Specifies whether the CA Single Sign-on Agent is enabled. Possible values are Yes and No. Default value is Yes. |
| AgentConfigObject | The Agent Configuration Object specified during installation. |
| SmHostFile | Path to the local Host Configuration File. Path can be specified in absolute terms or relative to *SMAGENT_HOME*.<br>**Note:** On Windows, you must specify paths using double backslashes ("\\") rather than single backslash ("\") to separate directories. On UNIX, use standard single slash ("/") separators. |

| Parameter | Description |
|---|---|
| | Example values:<br>(Windows) C:\\Program Files\\CA\\JBossAgent\\config\\SmHost.conf<br>(Windows) config\\SmHost.conf<br>(UNIX) export/JBossAgent/config/SmHost.conf<br>(UNIX) /config/SmHost.conf |
| ServerName | A string that will be used in the CA Single Sign-on Agent log to identify the JBoss Application Server. |
| appserverjaasloginhandler | Specifies the CA Single Sign-on Agent for JBoss login handler class. Default value is "com.ca.soa.agent.appserver.jaas.jboss.JBossLoginHandler". Do not change this value. |
| appserverjmshandler | Specifies the CA Single Sign-on Agent for JBoss JMS handler class. Default value is "com.ca.soa.agent.appserver.jaxrpc.jms.jboss.JBossJMSMessageHandler". Do not change this value. |

You should not need to edit the preconfigured values unless the location of the Host Configuration File changes or you want to refer to a different Agent Configuration Object. If you choose to use local configuration, you can add other Agent configuration parameters to these preconfigured values.

**Note:** Parameters held in the Agent configuration file are static; if you change these settings while the JBoss server is running, the CA Single Sign-on Agent will not pick up the change until JBoss is restarted.

The JavaAgent.conf file also contains a list of CA Single Sign-on Agent plugin classes; you do not need to alter this information.

Generally, you only need to edit the JavaAgent.conf.file if you change the name of your Agent Configuration Object.

**Sample JavaAgent.conf (Windows)**

```
# Java Agent Configuration File
#
# This file contains bootstrap information required by
# the SiteMinder Java Agent
#

#
# Configuration for agent testagent
#
defaultagentname=agentjboss
enablewebagent=yes
agentconfigobject=soaagentconfig
servername=jboss.example.com
smhostfile=C:\\Program Files\\CA\\JBossAgent\\config\\SmHost.conf

appserverjaasloginhandler=com.ca.soa.agent.appserver.jaas.jboss.JBossLoginHandler
appserverjmshandler=com.ca.soa.agent.appserver.jaxrpc.jms.jboss.JBossJMSMessageHandler

# Configure plugins for the agent testagent
transport_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.jaxrpcplugin.pluginconfig.JaxRpcPluginConfig, com.ca.soa.agent.jmsplugin.
pluginconfig.JMSPluginConfig, com.ca.soa.agent.jaxwsplugin.pluginconfig.
JaxWsPluginConfig
```

```
msg_body_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig, com.ca.
soa.agent.jaxwsplugin.pluginconfig.JaxWsPluginConfig
credential_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig, com.
ca.soa.agent.jaxwsplugin.pluginconfig.JaxWsPluginConfig
variable_resolver_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig

# <EOF>
```

## Agent Configuration Object

An Agent Configuration Object is a Policy Server object that holds Agent parameters for an Agent when using central agent configuration.

**Note:** Parameters held in an Agent Configuration Object are dynamic; if you change these settings while the JBoss server is running, the CA Single Sign-on Agent will pick up the change.

## CA Single Sign-on Agent Configuration Parameters

The following table contains a complete list of all Agent configuration parameters supported by the CA Single Sign-on Agent for JBoss.

Unless otherwise noted,you can define parameters in either the Agent Configuration Object or the Agent configuration file depending upon how you decide to configure the CA Single Sign-on Agent.

| Parameter Name | Value | Description |
|---|---|---|
| AcceptTPCookie | YES or NO | (Optional) If set to yes, configures the CA Single Sign-on Agent to assert identities from third-party CA Single Sign-on session cookies (that is, session cookies generated by custom Agents created using the CA Single Sign-on and CA Single Sign-On Web Services Security SDKs.<br>**Note:** AcceptTPCookie must be set to Yes to assert identities from session cookies generated by CA SOA Security Gateway.<br>Default is Yes. |
| AgentName | String | Defines the identity of the CA Single Sign-on Agent. It establishes a mapping between the name and the IP address of each web server instance hosting an Agent.<br>If a value is not set for this parameter, or if the CA Single Sign-on Agent does not find a match among the values listed, the CA Single Sign-on Agent uses the value set in the DefaultAgentName parameter instead.<br>**Note:** This parameter can have more than one value. Use the multi-value option when setting this parameter in an Agent Configuration Object. For local configuration files, add the parameter name followed by each value to separate lines in the file.<br>No default value. |
| AllowLocalConfig (Applies | YES or NO | If set to yes, parameters set locally in the Agent configuration file take precedence over parameters in the Agent Configuration Object.<br>Default is NO. |

| Parameter Name | Value | Description |
|---|---|---|
| only in the Agent Configuration Object) | | |
| AuthCacheSize | Number | (Optional) Size of the authentication cache for the CA Single Sign-on Agent (in number of entries). For example:<br>`authcachesize="1000"`<br><br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| AzCacheSize | Number | (Optional) Size of the authorization cache (in number of entries) for the CA Single Sign-on Agent. For example:<br>`authcachesize="1000"`<br><br>Default is 0.<br>To flush this cache, use the Administrative UI. |
| CacheTimeout | Number | (Optional) Number of seconds before cache times out. For example:<br>`cachetimeout="1000"`<br><br>Default is 600 (10 minutes). |
| ConfigObject (Applies only in Agent configuration file) | String | The name of the Agent Configuration Object associated with the CA Single Sign-on Agent.<br>No default value. |
| CookieDomain | String | (Optional) Name of the cookie domain. For example:<br>`cookiedomain="ca.com"`<br><br>No default value.<br>For more information, see the cookiedomainscope parameter. |
| CookieDomainScope | Number | (Optional) Further defines the cookie domain for assertion of CA Single Sign-on session cookies by the CA Single Sign-on Agent. The scope determines the number of sections, separated by periods, that make up the domain name. A domain always begins with a period (.) character. For example:<br>`cookiedomainscope="2"`<br><br>Default is 0, which takes the domain name specified in the cookiedomain parameter. |
| DefaultAgentName (Applies only in the Agent Configuration Object) | String | The agent identity the Policy Server will use to associate policies with the CA Single Sign-on Agent if there is no agent name specified in the AgentName parameter.<br>No default value. |

| | | |
|---|---|---|
| EnableWebAgent (Applies only in Agent configuration file) | YES or NO | Enables or disables the CA Single Sign-on Agent. When set to 'yes', the CA Single Sign-on Agent will protect resources using the Policies configured in the Policy Server for the configured agent identity.<br>Default is Yes. |
| LogOffUri | String | (Optional) The URI of a custom HTTP file that will perform a full log off (removing the session cookie from a user's browser). A fully qualified URI is not required. For example, LogOffUri could be set to: /Web pages/logoff.html<br>No default value. |
| PsPollInterval | Number | (Optional) The frequency with which the CA Single Sign-on Agent polls the Policy Server to retrieve information about policy changes.<br>Default is 30 seconds. |
| ResourceCacheSize | Number | (Optional) Size (in number of entries) of the cache for resource protection decisions. For example:<br>`resourcecachesize="1000"`<br>Default is 2000.<br>To flush this cache, use the Administrative UI. |
| SAMLSessionTicketLogoffi | YES or NO | (Optional) Determines whether the WSS Agent Security Interceptor should attempt to log off session tickets in SAML assertions.<br>Default is Yes. |
| ServerName (Applies only in Agent configuration file.) | String | A string to be used in the CA Single Sign-on Agent log to identify the target application server. |
| SessionGracePeriod | Number | (Optional) Grace period (in seconds) between the regeneration of session tokens.<br>Default is 30 |
| SmHostFile (Applies only in Agent configuration file) | String | Path to the local Host Configuration File (typically *SMAGENT_HOME*\conf\SmHost.conf).<br>No default value. |
| XMLAgentSoapFaultDetails | YES or NO | (Optional) Determines whether or not the WSS Agent Security Interceptor should insert the authentication/authorization rejection reason (if provided by the Policy Server) into the SOAP fault response sent to the web service consumer.<br>Default is No. |
| XMLSDKAcceptSMSessionCookie | YES or NO | (Optional) Determines whether or not the WSS Agent Security Interceptor accepts an CA CA Single Sign-on session cookie to authenticate a client.<br>Default is No. |

| Parameter Name | Value | Description |
|---|---|---|
| | | If set to Yes, the CA Single Sign-on Agent uses information in a session cookie sent as an HTTP header in the request as a means of authenticating the client. If set to No, session cookies are ignored and the CA Single Sign-on Agent requests credentials required by the configured authentication scheme. |
| XMLSDKMimeTypes | String | (Optional) A comma-delimited list of MIME types that the WSS Agent Security Interceptor will accept for processing by CA Single Sign-On Web Services Security. All POSTed requests having one of the listed MIME types are processed.<br>Examples:<br>text/xml<br>application/octet-stream<br>text/xml,multipart/related<br>If you do not add this parameter to the Agent Configuration Object, the WSS Agent Security Interceptor defaults to accepting text/xml and application /soap+xml MIME types. |

# Configure the Agent Environment to work with JBoss

For a JBoss 5.x or 6.x application server to work with the CA SSO Agent, you have to configure the environment. The following topics explain what needs to be configured.

- Configure Agent-related Environment Settings on JBoss 5.x (see page 965)
- Configure Agent-related Environment Settings on JBoss 6.x (see page 967)

## Configure Agent-related Environment Settings on JBoss 5.x

To configure the agent to operate with the JBoss 5.x Application Server, complete one of the following procedures:

- Set the JBoss 5.x Environment on Windows (see page 965).
- Set the JBoss 5.x Environment on UNIX (see page 966).

### Set the JBoss 5.x Environment on Windows

Before the Agent can operate with the JBoss Application Server, you must configure Agent-related environment settings on Windows by editing the JBoss run.bat script.

**To configure Agent-related environment settings**

1. Navigate to the *JBOSS_HOME*\bin directory
2. Open the run.bat file in a text editor.
3. Add the following entry to specify the installed location of the Agent for JBoss

   ```
   set SOA_HOME=SMAGENT_HOME
   ```

4. Add the following entry to define required JVM system properties for the agent:

```
set JAVA_OPTS=%JAVA_OPTS% -DJAVA_AGENT_ROOT=%SOA_HOME% -Dlog.log-config-
properties=%SOA_HOME%\config\log-config.properties -Dfile.encoding=UTF8
```

5. Add the following entry to include directories required for Agent operation in the JBOSS_CLASSPATH:

```
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%SOA_HOME%\config;%JBOSS_HOME%
\server\default\lib\cryptojFIPS.jar
```

6. By default, JBoss only listens for requests on the localhost IP address. To configure JBoss to listen on all IP addresses, locate the entry following the remark line "Execute the JVM in the background" and change "org.jboss.Main" to "org.jboss.Main -b 0.0.0.0". For example:

```
"%JAVA%" %JAVA_OPTS% -Djava.endorsed.dirs="%JBOSS_ENDORSED_DIRS%"
-classpath "%JBOSS_CLASSPATH%" org.jboss.Main -b 0.0.0.0 %*
```

7. Save your changes.

8. Restart the JBoss Application Server to apply the changes.

## Set the JBoss 5.x Environment on UNIX

Before the CA Single Sign-on Agent can operate with the JBoss Application Server, you must configure Agent-related environment settings on UNIX by editing the JBoss run.sh script.

**To configure CA Single Sign-on Agent-related environment settings**

1. Navigate to the *JBOSS_HOME*/bin directory

2. Open the run.sh file in a text editor.

3. Add the following lines to specify the installed location of the CA Single Sign-on Agent for JBoss:

```
SOA_HOME=SMAGENT_HOME
export SOA_HOME
```

4. Add the following entry to define required JVM system properties for the agent:

```
JAVA_OPTS=$JAVA_OPTS -DJAVA_AGENT_ROOT=$SOA_HOME -Dlog.log-config-
properties=$SOA_HOME/config/log-config.properties -Dfile.encoding=UTF8
export JAVA_OPTS
```

5. Add the following entry to include directories required for Agent operation in the JBOSS_CLASSPATH:

```
JBOSS_CLASSPATH=$JBOSS_CLASSPATH:$SOA_HOME/config:$JBOSS_HOME/server/default/lib
/cryptojFIPS.jar
export JBOSS_CLASSPATH
```

6. By default, JBoss only listens for requests on the localhost IP address. To configure JBoss to listen on all IP addresses, locate the entry following the remark line "Execute the JVM in the background" and change "org.jboss.Main" to "org.jboss.Main -b 0.0.0.0". For example:

```
"$JAVA" $JAVA_OPTS -Djava.endorsed.dirs="$JBOSS_ENDORSED_DIRS"
-classpath "$JBOSS_CLASSPATH" org.jboss.Main -b 0.0.0.0 *
```

7. Save your changes.

8. Restart the JBoss Application Server to apply the changes.

## Configure Agent-related Environment Settings on JBoss 6.x

To configure the agent to operate with a JBoss 6.x Application Server, complete one of the following procedures:

- Set the JBoss 6.x Environment on Windows (see page 967).
- Set the JBoss 6.x Environment on UNIX (see page 967).

### Set the JBoss 6.x Environment on Windows

Configure agent-related environment settings on Windows by editing the standalone.conf.bat script.

**Follow these steps:**

1. Navigate to the *JBOSS_HOME*\bin directory
2. Open the standalone.conf.bat file in a text editor.
3. Add the following entry to specify the installed location of the Agent for JBoss

   ```
   set SOA_HOME=SMAGENT_HOME
   ```

4. Add the *one* of the following entries to define required JVM system properties for the agent:

   **For JBoss versions before 6.4.5:**

   ```
   set "JAVA_OPTS=%JAVA_OPTS% -DJAVA_AGENT_ROOT=%SOA_HOME% -Dsmasa.home=%
   SOA_HOME%  -DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true -DTXM_DOCUMENT_BUILDER=org.
   apache.xerces.jaxp.DocumentBuilderFactoryImpl"
   ```

   **For JBoss version 6.4.5 and later:**

   ```
   set "JAVA_OPTS=%JAVA_OPTS% -DJAVA_AGENT_ROOT=%SOA_HOME% -Dsmasa.home=%SOA_HOME%
   -DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true -DTXM_DOCUMENT_BUILDER=org.apache.
   xerces.jaxp.DocumentBuilderFactoryImpl"
   ```

5. Save your changes.
6. Restart the JBoss Application Server to apply the changes.

### Set the JBoss 6.x Environment on UNIX

Configure agent-related environment settings on UNIX by editing the JBoss standalone.conf script.

**Follow these steps:**

1. Navigate to the *JBOSS_HOME*/bin directory
2. Open the standalone.conf file in a text editor.
3. Add the following lines to specify the installed location of the Agent for JBoss:

   ```
   SOA_HOME=SMAGENT_HOME
   export SOA_HOME
   ```

4. Add the *one* of the following entries to define required JVM system properties for the agent:

   **For JBoss versions before 6.4.5:**

   ```
   set "JAVA_OPTS=$JAVA_OPTS -DJAVA_AGENT_ROOT=$SOA_HOME -Dsmasa.home=$SOA_HOME  -
   DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true -DTXM_DOCUMENT_BUILDER=org.apache.xerces.
   jaxp.DocumentBuilderFactoryImpl"
   ```

   **For JBoss version 6.4.5 and later:**

   ```
   set "JAVA_OPTS=$JAVA_OPTS -DJAVA_AGENT_ROOT=$SOA_HOME -Dsmasa.home=$SOA_HOME -
   DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true -DTXM_DOCUMENT_BUILDER=org.apache.xerces.
   jaxp.DocumentBuilderFactoryImpl"
   ```

5. Save your changes.

6. Restart the JBoss Application Server to apply the changes.

# Configure CA SiteMinder® Agent for JBoss Logging

**Contents**



## Logging Overview

The CA Single Sign-on Agent for JBoss logger is implemented using Apache's log4j. For more information, see http://logging.apache.org/log4j/docs/.

Two log files provide important information about the CA Single Sign-on Agent:

- **CA Single Sign-on Agent logging**
  The agent writes information about its standard operations and performance such as error and processing messages to the CA Single Sign-on Agent log.

- **CA Single Sign-on Agent XML message processing logging file**
  In addition to its standard logging functionality, the agent also logs information relating specifically to WSS Agent Security Interceptor XML message processing.

> **Note:** CA Single Sign-on Agent XML message processing logging does not start until an XML message that needs to be processed is received.

## Configure CA Single Sign-on Agent Logging on JBoss 5.x

By default, CA Single Sign-on Agent logging is enabled and written to the XmlAgent.log file in one of the following locations:

- Windows—*JBOSS_HOME**bin*\soa-log\XmlAgent.log
- UNIX—*JBOSS_HOME*/bin/soa-log/XmlAgent.log

Change CA Single Sign-on Agent logging parameters by editing the log-config.properties file located in one of the following locations:

- Windows—*SMAGENT_HOME*\config\
- UNIX— *SMAGENT_HOME*/config/

**Note:** These are the default values; the logging configuration file name and location can be changed by editing the log.log-config-properties JVM system property.

Available logging parameters are as follows:

| Name | Description |
|---|---|
| log.logfile-append-on-reset | Add logging information to an existing log file instead of creating a new file each time logging is invoked.<br>Default value: no |
| log.logfile-pattern | Specifies the pathname (relative to *JBOSS_HOME*/bin) of the CA Single Sign-on Agent log file.<br>Default value: /soa-log/XmlAgent.log |
| log.logging-level | Defines the logging level. The levels are:<br>DEBUG - all logging, most verbose<br>CONFIG - configuration information<br>INFO - information<br>WARN -warnings<br>SEVERE - errors<br>Default value: SEVERE |
| log.logfile-limit | Specifies the size limit, in kilobytes Rollover a log file after it reaches the specified size.<br>Default value: 1000 |

## Configure CA Single Sign-on Agent XML Message Processing Logging on JBoss 5.x

By default, CA Single Sign-on Agent XML message processing logging is enabled and written to the soasm_agent.log file in one of the following locations:

- Windows—*SMAGENT_HOME*\bin\
- UNIX—*SMAGENT_HOME*/bin/

Change CA Single Sign-on Agent XML message processing logging parameters by editing the log.config file, located in one of the following directions:

- Windows—*SMAGENT_HOME*\config\

- UNIX— *SMAGENT_HOME*/config/

## Configure Logging on JBoss 6.x

To configure logging on JBoss 6.x, edit the standalone.xml file that is located in one of the following locations:

- **Windows**: *JBOSS_HOME*\standalone\configuration
- **UNIX**: *JBOSS_HOME*/standalone/configuration

In a text editor, add the following text to the logging subsystem section to configure logging with recommended default values:

```
<size-rotating-file-handler name="AgentFile" autoflush="true">
    <formatter>
      <pattern-formatter pattern="%d %-5p [%c] (%t) %m%n"/>
    </formatter>
    <file path="AGENT_HOME/log/XmlAgent.log"/>
    <rotate-size value="1000k"/>
    <max-backup-index value="5"/>
    <append value="true"/>
</size-rotating-file-handler>
<periodic-rotating-file-handler name="SOASMAgentFile" autoflush="true">
     <formatter>
          <pattern-formatter pattern="%d [%p] %c{3} %x - %m%n"/>
     </formatter>
     <file path="AGENT_HOME/log/soasm_agent.log"/>
       <suffix value=".yyyy-MM-dd"/>
          <append value="true"/>
  </periodic-rotating-file-handler>
  <logger category="com.ca.soa" use-parent-handlers="false">
    <level name="INFO"/>
    <handlers>
      <handler name="AgentFile"/>
    </handlers>
  </logger>
  <logger category="com.netegrity.tm" use-parent-handlers="false">
    <level name="INFO"/>
    <handlers>
      <handler name="SOASMAgentFile"/>
    </handlers>
  </logger>
```

You can change the values of the following configurable logging parameters:

- **<file path> (first instance)**
  Specifies the pathname of the CA Single Sign-on Agent log file.
- **<rotate-size value>**
  Specifies the size limit, in kilobytes before the CA Single Sign-on Agent log file rolls over.
- **<append value>**
  Specifies whether logging information is added to an existing log file instead of creating a file each time that logging is invoked. Specify one of the following values:
  - true
  - false

- **<file path> (second instance)**
  Specifies the pathname of the XML message processing log file.

- **<level name> (first instance)**
  Defines the CA Single Sign-on Agent logging level. Specify one of the following values:

  - ALL
  - TRACE
  - DEBUG
  - INFO
  - WARN
  - ERROR
  - FATAL
  - OFF

- **<level name> (second instance)**
  Defines the XML message processing logging level. Specify one of the following values:

  - ALL
  - TRACE
  - DEBUG
  - INFO
  - WARN
  - ERROR
  - FATAL
  - OFF

# Configure the CA SiteMinder® Agent for JBoss to Protect Web Applications

This section contains the following topics:

# Configure the SiteMinder Agent Security Interceptor to Protect Web Applications on JBoss 5.x

**Contents**



## Configure CA Single Sign-on Agent Authenticators

The Agent Authenticators extend the functionality of the JBossSX default authenticators with the ability to authenticate a user request based on an associated session cookie.

You can configure the Agent Authenticators into the JBoss security infrastructure for all web applications or for individual web applications as required.

## Configure Agent Authenticators For All Web Applications on JBoss 5.x

To configure the Agent Authenticators to handle all JBoss web application requests, replace the default JBossSX authenticator methods with the Agent Authenticator methods in the JBoss core authentication services definition.

The JBoss core authentication services are defined in the war-deployers-jboss-beans.xml configuration file located in the following location:

```
server/server_name/deployers/jbossweb.deployer/META-INF
```

**Note:** The Agent Authenticator methods extend the default authenticator methods; the default authenticator functionality is still available for requests without valid session cookies.

**To Configure the Agent Authenticators at the global level**

1. Navigate to server/*server_name*/deployers/jbossweb.deployer/META-INF.
2. Open the war-deployers-jboss-beans.xml file in a text editor.
3. Locate the <attribute name="Authenticators" ...> element definition section.
4. Edit the java:value element in the java:property element definitions for BASIC, FORM, CLIENT-CERT, and DIGEST authentication, replacing the default authenticator methods with the corresponding Agent Authenticator methods as required.

   To configure the SMJBossBasicAuthenticator, edit the java:property element for BASIC authentication as follows:

   ```
   <entry>
     <key>BASIC</key>
   ```

```
  <value>com.ca.soa.agent.appserver.authenticator.jBoss.
SMJBossBasicAuthenticator</value>
</entry>
```

To configure the SMJBossFormAuthenticator, edit the java:property element for FORM authentication as follows:

```
<entry>
  <key>FORM</key>
  <value>com.ca.soa.agent.appserver.authenticator.jBoss.
SMJBossFormAuthenticator</value>
</entry>
```

To configure the SMJBossClientCertAuthenticator, edit the java:property element for CLIENT-CERT authentication as follows:

```
<entry>
  <key>CLIENT-CERT</key>
  <value>com.ca.soa.agent.appserver.authenticator.jBoss.
SMJBossClientCertAuthenticator</value>
</entry>
```

To configure the SMJBossDigestAuthenticator, edit the java:property element for DIGEST authentication as follows:

```
<entry>
  <key>DIGEST</key>
  <value>com.ca.soa.agent.appserver.authenticator.jBoss.
SMJBossDigestAuthenticator</value>
</entry>
```

If you do not want the default authentication behavior to occur if session cookie validation fails, configure the SMJBossIdentityAsserter in place of any authenticator. For example, to configure the SMJBossIdentityAsserter so that default Digest authentication does not occur if identity assertion fails, edit the java:property element for DIGEST as follows:

```
<entry>
  <key>DIGEST</key>
  <value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossIdentityAsserter<
/value>
</entry>
```

**Note:** When configuring Agent Authenticators, the SMJBossClientCertAuthenticator and the SMJBossDigestAuthenticator require a session (SMSESSION cookie) to authenticate users. Without the SMSESSION cookie, X.509 certificate and digest authentication do not work.

5. Save the file and exit the text editor

The Agent Authenticators are configured as the default authenticators for all security-enabled web applications. The authenticator configured for the authentication method defined in the web application deployment descriptor will handle request unless an authenticator is configured individually for that application.

## Configure a Agent Authenticator for an Individual Application on JBoss 5.x

To configure a web application to use a specific Agent Authenticator to handle requests, define a context.xml file in the application WEB-INF directory. Configuring a context.xml file overrides the global authenticators defined in war-deployers-jboss-beans.xml.

**To configure a web application to use a specific Agent Authenticator**

1. Navigate to the application WEB-INF directory.

2. Open a text editor.

3. Define a context element containing a valve subelement that specifies the class name of the Agent Authenticator which you want to handle application requests.
   To configure the application to use SMJBossBasicAuthenticator, type:

   ```
   <Context cookies="true" crossContext="true">
     <Valve className="com.ca.soa.agent.appserver.authenticator.jBoss.
   SMJBossBasicAuthenticator"/>
   </Context>
   ```

   To configure the application to use the SMJBossFormAuthenticator, type:

   ```
   <Context cookies="true" crossContext="true">
     <Valve className="com.ca.soa.agent.appserver.authenticator.jBoss.
   SMJBossFormAuthenticator"/>
   </Context>
   ```

   To configure the application to use SMJBossClientCertAuthenticator, type:

   ```
   <Context cookies="true" crossContext="true">
     <Valve className="com.ca.soa.agent.appserver.authenticator.jBoss.
   SMJBossClientCertAuthenticator"/>
   </Context>
   ```

   To configure the application to use SMJBossDigestAuthenticator, type:

   ```
   <Context cookies="true" crossContext="true">
     <Valve className="com.ca.soa.agent.appserver.authenticator.jBoss.
   SMJBossDigestAuthenticator"/>
   </Context>
   ```

   To configure the application to use the SMJBossIdentityAsserter, type:

   ```
   <Context cookies="true" crossContext="true">
     <Valve className="com.ca.soa.agent.appserver.authenticator.jBoss.
   SMJBossIdentityAsserter"/>
   </Context>
   ```

   **Note:** When configuring Agent Authenticators, the SMJBossClientCertAuthenticator and the SMJBossDigestAuthenticator require a session (SMSESSION cookie) to authenticate users. Without the SMSESSION cookie, X.509 certificate and digest authentication do not work.

4. Save the file as context.xml and exit the text editor.

## Define a JBossSX Security Domain for the Agent Login Module

Define a JBoss security domain named SiteMinderDomain that configures the Agent Login Module required to authenticate credentials obtained by the Agent authenticators. Configure the SiteMinderDomain by adding an application-policy element to the login-config.xml file located in server/*server_name*/conf/.

**To configure Agent Authenticators at the global level**

1. Navigate to server/*server_name*/conf/login-config.xml
2. Open the login-config.xml file in a text editor.
3. Add the following application-policy element defining the SiteMinderDomain:

```
<application-policy name="SiteMinderDomain">
  <authentication>
    <login-module
    code="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossLoginModule"
    flag="required">
    <module-option name="unauthenticatedIdentity">anonymous</module-option>
    </login-module>
  </authentication>
</application-policy>
```

4. Save the file and exit the text editor.

## Configure Web Applications to Invoke the Agent Security Interceptor

To protect a web application using the CA SiteMinder Agent Security Interceptor, edit its deployment descriptor to enable security and map it to the SiteMinderDomain security domain.

## Edit the Application Deployment Descriptor to Enable Security

Edit the web.xml deployment descriptor to enable security for each web application that you want to protect with the Agent Web Interceptor. The web.xml file is located in the application WEB-INF directory.

For more information about the web.xml file and constituent element syntax, see the JBoss Enterprise Application Platform documentation.

**To Edit the web.xml deployment descriptor to enable security**

1. Navigate to the web application WEB-INF directory
2. Open the web.xml deployment descriptor file in a text editor.
3. Add one or more security-constraint elements defining what resources in the web application are to be protected. For example:

```
<security-constraint>
    <display-name>Constraint1</display-name>
    <web-resource-collection>
        <web-resource-name>admin resource</web-resource-name>
        <description/>
        <url-pattern>/admin/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
```

```
    <auth-constraint>
        <description/>
        <role-name>adminRole</role-name>
    </auth-constraint>
</security-constraint>
```

4. Add a security-role element defining roles used by the application. For example:

```
<security-role>
     <description/>
     <role-name>adminRole</role-name>
</security-role>
```

5. Add a login-config element. The auth-method subelement of the login-config element defines the authentication method (BASIC, FORMS, and so on) and therefore determines which globally configured Agent Authenticator will be invoked. For example, the following login-config element would result in the SMJBossFormAuthenticator handling application requests:

```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name/>
    <form-login-config>
        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/fail_login.jsp</form-error-page>
    </form-login-config>
</login-config>
```

6. Save the file and exit the text editor

7. Install or update the web application.

## Map Web Applications to the SiteMinderDomain Security Domain

Create a jboss-web.xml deployment descriptor file that defines the CA Single Sign-onDomain as the security domain for *each* web application that you want to protect with the CA Single Sign-on Agent. The jboss-web.xml file must be created in the application WEB-INF directory.

**To map a web application to the CA Single Sign-onDomain security domain**

1. Navigate to the application WEB-INF directory.

2. Open a text editor.

3. Enter the following:

```
<jboss-web>
<security-domain>java:/jaas/SiteMinderDomain</security-domain>
</jboss-web>
```

4. Save the file as jboss-web.xml and exit the text editor.

## Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the CA Single Sign-on Agent.

**To restart the JBoss Application Server**

1. If necessary, stop the JBoss Application Server process.

2. Open a command window.

3. Navigate to the *JBOSS_HOME*/bin directory.

4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the CA Single Sign-on Agent.

## Set Up the Agent Security Interceptor to Protect Web Applications on JBoss 6.x

**Contents**



### Configure the Agent Authenticator for Applications on JBoss 6.x

The Agent Authenticator extends the functionality of the JBossSX default authenticators with the ability to authenticate a user request that is based on an associated session cookie.

To configure a web application to use the Agent Authenticator to handle requests, create a jboss-web.xml file in the application WEB-INF directory. Configuring a jboss-web.xml file overrides the default authenticators.

**Follow these steps:**

1. Navigate to the application WEB-INF directory.

2. Open jboss-web.xml in a text editor.

3. Define a context element containing a valve subelement that specifies the class name of the Agent Authenticator which you want to handle application requests.
   To configure the application to use SMJBoss6BasicAuthenticator, type:

   ```
   <valve>
      <class-name>com.ca.soa.agent.appserver.authenticator.jboss.
   SMJBoss6BasicAuthenticator</class-name>
   </valve>
   ```

   To configure the application to use the SMJBoss6FormAuthenticator, type:

   ```
   <valve>
      <class-name>com.ca.soa.agent.appserver.authenticator.jboss.
   SMJBoss6FormAuthenticator</class-name>
   </valve>
   ```

   To configure the application to use SMJBoss6ClientCertAuthenticator, type:

```
<valve>
   <class-name>com.ca (http://com.ca).soa.agent.appserver.authenticator.jboss.
SMJBoss6ClientCertAuthenticator</class-name>
</valve>
```

To configure the application to use SMJBoss6DigestAuthenticator, type:

```
<valve>
   <class-name>com.ca.soa.agent.appserver.authenticator.jboss.
SMJBoss6DigestAuthenticator</class-name>
</valve>
```

To configure the application to use the SMJBoss6IdentityAsserter, type:

```
<valve>
   <class-name>com.ca.soa.agent.appserver.authenticator.jboss.
SMJBoss6IdentityAsserter</class-name>
</valve>
```

**Note:** When configuring Agent Authenticators, the SMJBossClientCertAuthenticator and the SMJBossDigestAuthenticator require a session (SMSESSION cookie) to authenticate users. Without the SMSESSION cookie, X.509 certificate and digest authentication do not work.

4. Save the file and exit the text editor.

## Make the Agent Java Class Accessible to Your Applications

To protect your applications with CA Single Sign-On, they must be able to access the Agent Java classes in module com.ca (http://com.ca).siteminder.jbossagent. To make the Agent Java classes accessible to your applications, do one of the following procedures:

- Configure the Agent as a Global Module
- Configure the Agent as an Application Dependency

## Configure the Agent as a Global Module

Configure the Agent as a global module by adding a new subsystem definition in the standalone.xml file.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. Add the following highlighted module name element to define the Agent as a global module in the "ee" web services subsystem:

```
<subsystem xmlns="urn:jboss:domain:ee:1.1">
  <global-modules>
<module name="com.ca.siteminder.jbossagent" slot="main"/>
  </global-modules>
  <spec-descriptor-property-replacement>false</spec-descriptor-property-
replacement>
  <jboss-descriptor-property-replacement>true</jboss-descriptor-property-
replacement>
</subsystem>
```

4. Save the file and exit the text editor.

**Notes:**

- To configure the WSS Agent JAX-WS HTTP Handler to protect all JAX-WS web services, you must add the Agent as a global module.
- Configuring the Agent as a global module makes it accessible to all web applications (using the Agent Security Interceptor) and web services (using the WSS Agent Security Interceptor).
- There is a conflict between the default JBoss and CA Single Sign-On xml-security libraries. If you configure the Agent as a global module, you must resolve that conflict.

## Resolve a Conflict Between the JBoss and WSS Agent Security Libraries

**Only complete this procedure for CA Single Sign-On versions 12.52 SP1 CR05 and earlier.**

There is a conflict between the default JBoss and CA Single Sign-On XML security library (org.apache.santuario.xmlsec). If you configure the Agent as a global module, remove the security library from the module definitions in the module.xml file.

**Follow these steps:**

1. Navigate to the following location:
   - **Windows:** *JBOSS_HOME*\modules\system\layers\base\org\jboss\as\webservices\server\integration\main
   - **UNIX:** *JBOSS_HOME*/modules/system/layers/base/org/jboss/as/webservices/server/integration/main
2. Open the **module.xml** file in a text editor.
3. Locate the security library entry (org.apache.santuario.xmlsec) and comment out the following line:

   ```
   <!-- <module name="org.apache.santuario.xmlsec" export="true"/> -->
   ```

For applications that depend on the default JBoss XML Security library, do one of the following procedures to enable them to access to it:

- Package the org.apache.santuario.xmlsec JAR files as a separate module from the JBoss web services module and configure it as a dependency for those applications.
- Include the org.apache.santuario.xmlsec JAR files in the application WAR file.

## Configure the Agent as a Per-Application Dependency

If the JBoss Agent is not defined as a global module, define it as a dependency in the jboss-deployment-structure.xml file of each application that you want to protect.

**Follow these steps for JBoss 6.x:**

1. Navigate to the application WEB-INF directory.
2. Open jboss-deployment-structure.xml in a text editor.
3. Add the following module name element to the dependencies element:

   ```
   <module name="com.ca (http://com.ca).siteminder.jbossagent" />
   ```

   For example:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <jboss-deployment-structure>
      <deployment>
          <dependencies>
                <module name="com.ca (http://com.ca).siteminder.jbossagent" />
          </dependencies>
      </deployment>
   </jboss-deployment-structure>
   ```

4. Save the file and exit the text editor.

## Define a JBossSX Security Domain for the Agent Security Interceptor on JBoss 6.x

For the Agent security interceptor, define a JBossSX security domain that is named SiteMinderDomain by adding a <security-domain-name> element to the standalone.xml file.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows:** *JBOSS_HOME*\standalone\configuration
   - **UNIX:** *JBOSS_HOME*/standalone/configuration
2. Open the standalone.xml file in a text editor.
3. Add *one* of the following <security-domain-name> elements:
   **For JBoss versions before 6.4.5:**

   ```
   <security-domain name="SiteMinderDomain" cache-type="default">
      <authentication>
         <login-module code="com.ca (http://com.ca).soa.agent.appserver.
   authenticator.jBoss.SMJBossLoginModule" flag="required"/>
      </authentication>
   </security-domain>
   ```

   **For JBoss version 6.4.5 and later:**

   ```
   <security-domain name="SiteMinderDomain" cache-type="default">
      <authentication>
         <login-module code="com.ca (http://com.ca).soa.agent.appserver.
   authenticator.jBoss.SMJBossLoginModule" flag="required" module="com.ca (http://co
   m.ca/).siteminder.jbossagent"/>
   ```

```
   </authentication>
</security-domain>
```

4. Save the file and exit the text editor.

## Configure Web Applications to Invoke the Agent Security Interceptor

To protect a web application using the Agent Security Interceptor, edit its deployment descriptor to enable security and map it to the SiteMinderDomain security domain.

## Edit the Application Deployment Descriptor to Enable Security

Edit the web.xml deployment descriptor to enable security for each web application that you want to protect with the Agent Web Interceptor. The web.xml file is located in the application WEB-INF directory.

For more information about the web.xml file and constituent element syntax, see the JBoss Enterprise Application Platform documentation.

**Follow these steps:**

1. Navigate to the web application WEB-INF directory

2. Open the web.xml deployment descriptor file in a text editor.

3. Add one or more security-constraint elements defining what resources in the web application are to be protected. For example:

```
<security-constraint>
    <display-name>Constraint1</display-name>
    <web-resource-collection>
        <web-resource-name>admin resource</web-resource-name>
        <description/>
        <url-pattern>/admin/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>adminRole</role-name>
    </auth-constraint>
</security-constraint>
```

4. Add a security-role element defining roles that are used by the application. For example:

```
<security-role>
     <description/>
     <role-name>adminRole</role-name>
</security-role>
```

5. Add a login-config element. The auth-method subelement of the login-config element defines the authentication method (BASIC, FORMS, and so on) and therefore determines which globally configured Agent Authenticator is invoked. For example, the following login-config element would result in the SMJBossFormAuthenticator handling application requests:

```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name/>
    <form-login-config>
        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/fail_login.jsp</form-error-page>
```

```
    </form-login-config>
</login-config>
```

6. Save the file and exit the text editor

7. Install or update the web application.

### Map Web Applications to the SiteMinderDomain Security Domain

Create a jboss-web.xml deployment descriptor file that defines the SiteMinderDomain as the security domain for *each* web application that you want to protect with the Agent. The jboss-web.xml file must be created in the application WEB-INF directory.

**Follow these steps:**

1. Navigate to the application WEB-INF directory.

2. Open a text editor.

3. Enter the following code:

```
<jboss-web>
  <security-domain>java:/jaas/SiteMinderDomain</security-domain>
</jboss-web>
```

4. Save the file as jboss-web.xml and exit the text editor.

### Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the CA Single Sign-On Agent.

**To restart the JBoss Application Server**

1. If necessary, stop the JBoss Application Server process.

2. Open a command window.

3. Navigate to the JBOSS_HOME/bin directory.

4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the CA Single Sign-On Agent.

## Configure SiteMinder Policies to Protect JBoss Web Applications

**Contents**

- Configure Security Policies for the Proxy Server Web Agent (see page 986)

## Configure a CA Single Sign-on Agent Security Interceptor Authentication Realm

Configure an authentication realm on the Policy Server to allow the CA Single Sign-on Agent Security Interceptor to validate users credentials using information obtained from CA Single Sign-On session cookies. Use the Administrative UI to create the CA Single Sign-on Agent Security Interceptor authentication realm.

**Follow these steps:**

1. Click Policies, Domains.
2. Click Domain, Create Domain.
3. The Create Domain pane opens.

> **Note:** You can click Help for a description of fields, controls, and their respective requirements.

4. Type the name and a description of the Domain in the fields on the General group box.
5. Add one or more user directories that contain the users who can access the protected resources.
6. Create the validation realm:
    a. Click the Realms tab on the Domain pane, New Realm, OK.
    b. The Create Realm pane opens.
    c. Enter the following information:
        - Name: A unique name for the realm—for example, CA Single Sign-on Agent Security Interceptor Validation Realm
        - Description: An optional description for the validation realm
        - Agent: The name of the agent identity that you created for the Agent for JBoss.
        - Resource Filter: /smauthenticationrealm
        - Authentication Scheme: Basic

> **Note:** You do not need to configure any rules for the validation realm.

d. Specify session properties in the Session group box:

- Disable all session time-outs
- Ensure the No Persistent Session option is selected

e. Click Finish.
The Create Realm Task is submitted for processing.

7. Click Submit.
The Create Domain Task is submitted for processing.

## (Optional) Configure the Agent to Return Group Membership to JBoss Using Responses

The CA Single Sign-on Agent Web Interceptor can be configured to return physical or virtual group membership information to JBoss using CA Single Sign-on HTTP header responses from the Policy Server during user authentication.

When the CA Single Sign-on Agent Web Interceptor receives responses containing the _SM_JBOSS_GROUP=*group name* syntax, the CA Single Sign-on Agent Web Interceptor converts the *group_name* value to a J2EE principal and adds this principal to the subject after successful authentication.

- ***group_name***
Specifies a response attribute value from the Policy Server that could be a physical group name from the user store or a virtual group.

The CA Single Sign-on Agent adds the same amount of group principals as responses received from the Policy Server.

**Note:** The CA Single Sign-on Agent Web Interceptor can only process _SM_JBOSS_GROUP response attributes to return group membership information to JBoss. It cannot process other response attributes added to HTTP header variables to pass information to a web application.

**To configure Groups as responses for the CA Single Sign-on Agent**

1. Configure an OnAuthAccept group authentication rule with a * resource filter in the CA Single Sign-on Authentication Realm.

2. Create CA Single Sign-on HTTP header responses using the _SM_JBOSS_GROUP variable name in the policy domain for the CA Single Sign-on Authentication Realm.

**Note:** The CA Single Sign-on Administrative UI shows an additional underscore before "_SM_JBOSS_GROUP" when it displays the variable name, so that it appears as "HTTP__SM_JBOSS_GROUP". This is not an error and can be ignored.

3. In the policy domain for the CA Single Sign-on Authentication Realm:

    a. Create a group policy.

    b. Attach the users who belong to the group policy.

    c. Attach the group authentication rule to this policy.

    d. Bind the group response to the group authentication rule.

## Example Configure the CA Single Sign-on Agent Web Interceptor to return groups using responses

The following example shows one method of configuring the CA Single Sign-on Agent Web Interceptor to return groups using responses:

1. In the CA Single Sign-on Authentication Realm, configure an OnAuthAccept rule named **Group Authentication Rule** with a ***** resource filter.

2. In the policy domain for the CA Single Sign-on Authentication Realm, create CA Single Sign-on responses with a static HTTP header attribute for the following sample JBoss groups:

    - **Group Administrators**
      Attribute kind: Static HTTP Header
      Variable name: _SM_JBOSS_GROUP
      Variable value: Administrators

    - **Group Deployers**
      Attribute kind: Static HTTP Header
      Variable name: _SM_JBOSS_GROUP
      Variable value: Deployers

    - **Group Monitors**
      Attribute kind: Static HTTP Header
      Variable name: _SM_JBOSS_GROUP
      Variable value: Monitors

    - **Group Operators**
      Attribute kind: Static HTTP Header
      Variable name: _SM_JBOSS_GROUP
      Variable value: Operators

3. In the policy domain for the CA Single Sign-on Authentication Realm:

    a. Configure a policy named **Group Administrator Policy**.

    b. Attach the Administrator group or users, who belong to the Administrator group, to this policy.

    c. Attach the Group Authentication Rule to this policy.

    d. Bind the Group Administrator response to this rule.

e. Repeat this step and configure separate policies for the Deployers, Operators, and Monitors groups.

f. Bind the Group Administrator response to this rule.

4. Repeat Step 3 to configure separate policies for the Deployers, Operators, and Monitors groups.

### Configure Security Policies for the Proxy Server Web Agent

To configure the CA Single Sign-on Agent for JBoss to protect web applications by perimeter authentication, create policies that specify how the Web Agent on the proxy server controls access to the URL that represents the proxied JBoss web application resources.

# Configure the CA SiteMinder® Agent for JBoss to Protect Web Services

This section contains the following topics:



## Configure the WSS Agent Security Interceptor to Protect Web Services on JBoss 5.x

**Contents**



### Configure WSS Agent Security Interceptor Protection for JAX-RPC Web Services Over HTTP Transport

To configure the WSS Agent Web Interceptor to protect JAX-RPC web services over HTTP transport, configure those services to invoke the WSS Agent JAX-RPC HTTP Handler. You can configure global use of the JAX-RPC Handler for all JAX-RPC HTTP web services or configure it for individual web services, as required.

### Configure the WSS Agent JAX-RPC HTTP Handler for all JAX-RPC HTTP Web Services

To configure the WSS Agent JAX-RPC Handler to be invoked for all JAX-RPC HTTP web services, add the WSS Agent JAX-RPC Handler class (com.ca.soa.agent.jaxrpcplugin.JaxrpcHandler) to the standard JAX-RPC endpoint configuration file, standard-jaxrpc-endpoint-config.xml.

By default, the standard-jaxrpc-endpoint-config.xml file is in the following location:

*JBOSS_HOME*/server/*instance_type*/deployers/jbossws.deployer/META-INF

- ***instance_type***
  Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

**Follow these steps:**

1. Navigate to the location of the standard-jaxrpc-endpoint-config.xml file for your JBoss version and instance type.

2. Open the standard-jaxrpc-endpoint-config.xml file in a text editor.

3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined.

```
<handler>
   <j2ee:handler-name>SM XMLAgentJaxrpc Handler</j2ee:handler-name>
   <j2ee:handler-class>
     com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler
   </j2ee:handler-class>
</handler>
```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-RPC Handler will be invoked for all JAX-RPC web services.

**Example standard-jaxrpc-endpoint-config.xml file**

```
<jaxrpc-config xmlns="urn:jboss:jaxrpc-config:2.0" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xmlns:j2ee="http://java.sun.com/xml/ns/j2ee"
xsi:schemaLocation="urn:jboss:jaxrpc-config:2.0 jaxrpc-config_2_0.xsd">

   <endpoint-config>
     <config-name>Standard Endpoint</config-name>
       <pre-handler-chain>
         <handler-chain-name>SM XMLAgentJaxrpc Handlers</handler-chain-name>
         <handler>
           <j2ee:handler-name>SM XMLAgentJaxrpc Handler</j2ee:handler-name>
           <j2ee:handler-class>
             com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler
           </j2ee:handler-class>
         </handler>
       </pre-handler-chain>
  </endpoint-config>

</jaxrpc-config>
```

## Configure the WSS Agent JAX-RPC HTTP Handler for a Single Web Service

Configure individual JAX-RPC HTTP web services to invoke the WSS Agent JAX-RPC HTTP Handler by defining the com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler in the application webservices.xml deployment descriptor.

For example:

```
<webservices ...>
   <webservice-description>
    ...
```

```
    <port-component>
      ...
      <handler>
        <handler-name>SM XMLAgentJaxrpc Handler</handler-name>
        <handler-class>com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler<
/handler-class>
      </handler>
    </port-component>
   </webservice-description>
  </webservices>
```

The JBoss WSS Agent JAX-RPC HTTP Handler will be invoked only for this web service.

## Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over HTTP Transport

To configure the WSS Agent Security Interceptor to protect JAX-WS web services over HTTP transport, configure those services to invoke the WSS Agent JAX-WS HTTP Handler. You can configure global use of the JAX-WS Handler for all JAX-WS HTTP web services or configure it for individual web services, as required.

## Configure the WSS Agent JAX-WS HTTP Handler for all JAX-WS HTTP Web Services

To configure the WSS Agent JAX-WS HTTP Handler to be invoked for all JAX-WS HTTP web services, add the WSS Agent JAX-WS Handler class (com.ca.soa.agent.jaxwsplugin.JaxWsHandler) to the standard JAX-WS endpoint configuration file, standard-jaxws-endpoint-config.xml.

By default, the standard-jaxws-endpoint-config.xml file is in the following location:

*JBOSS_HOME*/server/*instance_type*/deployers/jbossws.deployer/META-INF

- ***instance_type***
  Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

**Follow these steps:**

1. Navigate to the location of the standard-jaxws-endpoint-config.xml file for your JBoss version and instance type.

2. Open the standard-jaxws-endpoint-config.xml file in a text editor.

3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined:

   ```
   <javaee:handler>
      <javaee:handler-name>
         JBoss JAX-WS PEP Interceptor
      </javaee:handler-name>
      <javaee:handler-class>
         com.ca.soa.agent.jaxwsplugin.JaxWsHandler
      </javaee:handler-class>
   </javaee:handler>
   ```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler will be invoked for all JAX-WS web services.

**Example standard-jaxws-endpoint-config.xml file**

```
<jaxws-config xmlns="urn:jboss:jaxws-config:2.0" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xmlns:javaee="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="urn:jboss:jaxws-config:2.0 schema/jaxws-config_2_0.xsd">

<endpoint-config>
    <config-name>Standard Endpoint</config-name>
    <pre-handler-chains>
      <javaee:handler-chain>
        <javaee:protocol-bindings>##SOAP11_HTTP</javaee:protocol-bindings>

        <javaee:handler>
           <javaee:handler-name>
            JBoss JAX-WS PEP Interceptor
           </javaee:handler-name>
           <javaee:handler-class>
            com.ca.soa.agent.jaxwsplugin.JaxWsHandler
           </javaee:handler-class>
        </javaee:handler>

        <javaee:handler>
          <javaee:handler-name>Recording Handler</javaee:handler-name>
          <javaee:handler-class>
            org.jboss.wsf.framework.invocation.RecordingServerHandler
          </javaee:handler-class>
        </javaee:handler>

      </javaee:handler-chain>
    </pre-handler-chains>
</endpoint-config>
```

## Configure the WSS Agent JAX-WS HTTP Handler for a Single JAX-WS HTTP Web Service

Configure individual JAX-WS HTTP web services to invoke the WSS Agent JAX-WS Handler.

**Follow these steps:**

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
   <handler-chain>
   <handler>
   <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>
   <handler-class>com.ca.soa.agent.jaxwsplugin.JaxWsHandler</handler-class>
   </handler>
   </handler-chain>
   </handler-chains>
   ```

2. Add the following JWS annotation to the web service JWS file:

   ```
   @HandlerChain(file = "Services_handler.xml")
   ```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service.

The JBoss WSS Agent JAX-WS Handler is invoked for the web service.

## Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over JMS Transport

To configure the WSS Agent Web Interceptor to protect JAX-WS web services over JMS transport, configure those services to invoke the WSS Agent JAX-WS JMS Handler. You can configure global use of the JAX-WS JMS Handler for all JAX-WS JMS web services or configure it for individual web services, as required.

**Important!** Do not place the WSS Agent JAX-WS HTTP Handler and the WSS Agent JAX-WS JMS Handler in the same handler chain. If you configure either handler in the default handler chain for the container, verify that all JAX-WS web services in the container use the corresponding transport.

## Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services

To configure the WSS Agent JAX-WS JMS Handler to be invoked for all JAX-WS JMS web services, add the WSS Agent JAX-WS JMS Handler class (com.ca.soa.agent.jmsplugin.JaxWsJMSHandler) to the standard JAX-WS endpoint configuration file, standard-jaxws-endpoint-config.xml.

The standard-jaxws-endpoint-config.xml file is located in *JBOSS_HOME*/server/*instance_type* /deployers/jbossws.deployer/META-INF.

- ***instance_type***
  Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

**Follow these steps:**

1. Navigate to *JBOSS_HOME*/server/default/deployers/jbossws.deployer/META-INF.
2. Open the standard-jaxws-endpoint-config.xml file in a text editor.
3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined:

```
<javaee:handler>
   <javaee:handler-name>
      JBoss JAX-WS PEP Interceptor
   </javaee:handler-name>
   <javaee:handler-class>
      com.ca.soa.agent.jmsplugin.JaxWsJMSHandler
   </javaee:handler-class>
</javaee:handler>
```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler will be invoked for all JAX-WS web services.

**Example standard-jaxws-endpoint-config.xml file**

```
<jaxws-config xmlns="urn:jboss:jaxws-config:2.0" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xmlns:javaee="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="urn:jboss:jaxws-config:2.0 schema/jaxws-config_2_0.xsd">

<endpoint-config>
    <config-name>Standard Endpoint</config-name>
    <pre-handler-chains>
      <javaee:handler-chain>
        <javaee:protocol-bindings>##SOAP11_HTTP</javaee:protocol-bindings>

        <javaee:handler>
           <javaee:handler-name>
            JBoss JAX-WS PEP Interceptor
           </javaee:handler-name>
           <javaee:handler-class>
```

```
            com.ca.soa.agent.jmsplugin.JaxWsJMSHandler
          </javaee:handler-class>
        </javaee:handler>

        <javaee:handler>
          <javaee:handler-name>Recording Handler</javaee:handler-name>
          <javaee:handler-class>
            org.jboss.wsf.framework.invocation.RecordingServerHandler
          </javaee:handler-class>
        </javaee:handler>

      </javaee:handler-chain>
    </pre-handler-chains>
</endpoint-config>
```

## Configure the WSS Agent JAX-WS Handler for a Single JAX-WS JMS Web Service

You can configure individual JAX-WS JMS web services to invoke the WSS Agent JAX-WS JMS Handler.

**Follow these steps:**

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

```
<?xml version="1.0" encoding="UTF-8"?>
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>
      <handler-class>com.ca.soa.agent.jmsplugin.JaxWsJMSHandler</handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

2. Add the following JWS annotation to the web service JWS file:

```
@HandlerChain(file = "Services_handler.xml")
```

The JBoss WSS Agent JAX-WS Handler will be invoked only for this web service.

## Configure the WSS Agent Login Module

Define a JBoss security domain named system.XMLAgent that configures the WSS Agent Login Module required to authenticate credentials obtained by the WSS Agent Handlers.

You configure the system.XMLAgent by adding an application-policy element to the login-config.xml file located in *JBOSS_HOME*/server/*instance_type*/conf.

- ***instance_type***
  Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

**To configure CA Single Sign-on Agent Authenticators at the global level**

1. Navigate to server/*server_name*/conf/
2. Open the login-config.xml file in a text editor.
3. Add the following application-policy element defining the CA Single Sign-onDomain:

```
<application-policy name="system.XMLAgent">
    <authentication>
      <login-module code="com.ca.soa.agent.appserver.jaas.XMLAgentLoginModule"
        flag="required">
        <module-option name="unauthenticatedIdentity">anonymous</module-option>
      </login-module>
    </authentication>
</application-policy>
```

4. Save the file and exit the text editor.

### Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the CA Single Sign-on Agent.

**To restart the JBoss Application Server**

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the CA Single Sign-on Agent.

## Configure the WSS Agent Security Interceptor to Protect Web Services on JBoss 6.x

**Contents**



### Make the CA Single Sign-On Agent Java Class Accessible to Your Applications

To protect your applications with CA Single Sign-On, they must be able to access the Agent Java classes in module com.ca (http://com.ca).siteminder.jbossagent. To make the Agent Java classes accessible to your applications, do one of the following procedures:

- Configure the Agent as a Global Module
- Configure the Agent as an Application Dependency

## Configure the Agent as a Global Module

Configure the Agent as a global module by adding a new subsystem definition in the standalone.xml file.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. Add the following highlighted module name element to define the Agent as a global module in the "ee" web services subsystem:

```
<subsystem xmlns="urn:jboss:domain:ee:1.1">
  <global-modules>
<module name="com.ca.siteminder.jbossagent" slot="main"/>
  </global-modules>
  <spec-descriptor-property-replacement>false</spec-descriptor-property-
replacement>
  <jboss-descriptor-property-replacement>true</jboss-descriptor-property-
replacement>
</subsystem>
```

4. Save the file and exit the text editor.

**Notes:**

- To configure the WSS Agent JAX-WS HTTP Handler to protect all JAX-WS web services, you must add the Agent as a global module.
- Configuring the Agent as a global module makes it accessible to all web applications (using the Agent Security Interceptor) and web services (using the WSS Agent Security Interceptor).
- There is a conflict between the default JBoss and CA Single Sign-On xml-security libraries. If you configure the Agent as a global module, you must resolve that conflict.

**For CA Single Sign-On versions 12.52 SP1 CR05 and earlier:** If the Agent is defined as a global module, there is a conflict between the default JBoss and CA Single Sign-On XML security library (org. apache.santuario.xmlsec). If you configure the Agent as a global module, remove the security library from the module definitions in the module.xml file.

**Follow these steps:**

1. Navigate to the following location:
   - **Windows:** *JBOSS_HOME*\ modules\system\layers\base\org\jboss\as\webservices\server\integration\main
   - **UNIX:** *JBOSS_HOME*/modules/system/layers/base/org/jboss/as/webservices/server /integration/main

2. Open the **module.xml** file in a text editor.

3. Locate the security library entry (org.apache.santuario.xmlsec) and comment out the following line:

   ```
   <!-- <module name="org.apache.santuario.xmlsec" export="true"/> -->
   ```

For applications that depend on the default JBoss XML Security library, do one of the following procedures to enable them to access to it:

- Package the org.apache.santuario.xmlsec JAR files as a separate module from the JBoss web services module and configure it as a dependency for those applications.
- Include the org.apache.santuario.xmlsec JAR files in the application WAR file.

## Configure the Agent as a Per-Application Dependency

If the JBoss Agent is not defined as a global module, define it as a dependency in the jboss-deployment-structure.xml file of each application that you want to protect.

**Follow these steps:**

1. Navigate to the application WEB-INF directory.

2. Open jboss-deployment-structure.xml in a text editor.

3. Add the following module name element to the dependencies element:

   ```
   <module name="com.ca (http://com.ca).siteminder.jbossagent" />
   ```

   For example:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <jboss-deployment-structure>
      <deployment>
          <dependencies>
                <module name="com.ca (http://com.ca).siteminder.jbossagent" />
          </dependencies>
      </deployment>
   </jboss-deployment-structure>
   ```

4. Save the file and exit the text editor.

## Configure the WSS Agent JAX-RPC HTTP Handler to Protect Web Services in JBoss 6.x

Configure each JAX-RPC HTTP web service to invoke the WSS Agent JAX-RPC HTTP Handler.

**Note**: There is no global way to configure the WSS Agent JAX-RPC HTTP Handler to protect all JAX_RPC web services.

**Follow these steps:**

1. Open the application webservices.xml deployment descriptor in a text editor.

2. Define the com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler.
   For example:

```
<webservices ...>
   <webservice-description>
    ...
    <port-component>
      ...
      <handler>
        <handler-name>SM XMLAgentJaxrpc Handler</handler-name>
        <handler-class>com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler<
/handler-class>
      </handler>
    </port-component>
   </webservice-description>
  </webservices>
```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service (see page )
   .

The JBoss WSS Agent JAX-RPC HTTP Handler is invoked only for this web service.

## Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over HTTP Transport

To configure the WSS Agent Security Interceptor to protect JAX-WS web services over HTTP transport, configure those services to invoke the WSS Agent JAX-WS HTTP Handler. You can configure global use of the JAX-WS Handler for all JAX-WS HTTP web services or configure it for individual web services, as required.

## Configure the WSS Agent JAX-WS HTTP Handler to Protect all JAX-WS HTTP Web Services on JBoss 6.x

To configure the WSS Agent Security Interceptor to protect all JAX-WS HTTP web services, make the following changes to standalone.xml:

- Add a subsystem definition to configure the agent as a global module (if it is not already present).
- Add a pre-handler-chain definition to configure the WSS Agent JAX-WS HTTP Handler as the handler for all JAX-WS HTTP web services.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone-full.xml in a text editor.
3. If it is not already defined, configure the Agent as a global module
4. Add the following pre-handler-chain element to the "Standard Endpoint" endpoint-config element in the web services subsystem definition as the first such element defined:

```
<pre-handler-chain name="WSSAgent" protocol-bindings="
##SOAP11_HTTP ##SOAP12_HTTP">
        <handler name="SoaJaxWsHandler" class="com.ca.soa.agent.jaxwsplugin.
```

```
JaxWsHandler"/>
 </pre-handler-chain>
```

**Note**: The default standalone-full.xml does not have a web services subsystem predefined. If no web services subsystem is present, add one that includes the previous pre-handler-chain element. For example:

```
<subsystem xmlns="urn:jboss:4domain:webservices:1.2">
  <modify-wsdl-address>true</modify-wsdl-address>
  <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
  <endpoint-config name="Standard-Endpoint-Config">
     <pre-handler-chain name="WSSAgent" protocol-bindings="
##SOAP11_HTTP ##SOAP12_HTTP">
        <handler name="SoaJaxWsHandler" class="com.ca.soa.agent.jaxwsplugin.
JaxWsHandler"/>
     </pre-handler-chain>
  </endpoint-config>
  <client-config name="Standard-Client-Config"/>
</subsystem>
```

5. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler is invoked for all JAX-WS HTTP web services.

## Configure the WSS Agent JAX-WS HTTP Handler for a Single JAX-WS HTTP Web Service

Configure individual JAX-WS HTTP web services to invoke the WSS Agent JAX-WS Handler.

**Follow these steps:**

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

```
<?xml version="1.0" encoding="UTF-8"?>
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
<handler-chain>
<handler>
<handler-name>JBoss JAX-WS PEP Interceptor</handler-name>
<handler-class>com.ca.soa.agent.jaxwsplugin.JaxWsHandler</handler-class>
</handler>
</handler-chain>
</handler-chains>
```

2. Add the following JWS annotation to the web service JWS file:

```
@HandlerChain(file = "Services_handler.xml")
```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service.

The JBoss WSS Agent JAX-WS Handler is invoked for the web service.

## Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over JMS Transport on JBoss 6.x

To configure the WSS Agent Web Interceptor to protect JAX-WS web services over JMS transport, configure those services to invoke the WSS Agent JAX-WS JMS Handler. You can configure global use of the JAX-WS JMS Handler for all JAX-WS JMS web services or configure it for individual web services, as required.

**Important!** Do not place the WSS Agent JAX-WS HTTP Handler and the WSS Agent JAX-WS JMS Handler in the same handler chain. If you configure either handler in the default handler chain for the container, verify that all JAX-WS web services in the container use the corresponding transport.

## Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services on JBoss 6.x

To configure the WSS Agent Security Interceptor to protect all JAX-WS JMS web services, make the following changes to standalone.xml:

- Add a subsystem definition to configure the agent as a global module (if it is not already present).
- Add a pre-handler-chain definition to configure the WSS Agent JAX-WS JMS Handler as the handler for all JAX-WS JMS web services.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. If it is not already defined, configure the Agent as a global module
4. Add the following pre-handler-chain element to the "Standard Endpoint" endpoint-config element in the web services subsystem definition as the first such element defined:

```
<pre-handler-chain name="WSSAgent" protocol-bindings="
##SOAP11_HTTP ##SOAP12_HTTP">
        <handler name="SoaJaxWsJMSHandler" class="com.ca.soa.agent.jaxwsplugin.
JaxWsJMSHandler"/>
 </pre-handler-chain>
```

The default standalone.xml does not have a web services subsystem predefined. If no web services subsystem is present, add one that includes the previous pre-handler-chain element. For example:

```
<subsystem xmlns="urn:jboss:4domain:webservices:1.2">
  <modify-wsdl-address>true</modify-wsdl-address>
  <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
  <endpoint-config name="Standard-Endpoint-Config">
```

```
     <pre-handler-chain name="WSSAgent" protocol-bindings="
##SOAP11_HTTP ##SOAP12_HTTP">
        <handler name="SoaJaxWsJMSHandler" class="com.ca.soa.agent.jaxwsplugin.
JaxWsJMSHandler"/>
     </pre-handler-chain>
  </endpoint-config>
  <client-config name="Standard-Client-Config"/>
</subsystem>
```

5. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler is invoked for all JAX-WS JMS web services.

## Configure the WSS Agent JAX-WS Handler for a Single JAX-WS JMS Web Service on JBoss 6.x

You can configure individual JAX-WS JMS web services to invoke the WSS Agent JAX-WS JMS Handler.

**Follow these steps:**

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

```
<?xml version="1.0" encoding="UTF-8"?>
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>
      <handler-class>com.ca.soa.agent.jmsplugin.JaxWsJMSHandler</handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

2. Add the following JWS annotation to the web service JWS file:

```
@HandlerChain(file = "Services_handler.xml")
```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service.

The JBoss WSS Agent JAX-WS Handler is invoked only for this web service.

## Define a JBossSX Security Domain for the Agent Login Module on JBoss 6.x

Define a JBoss security domain named SiteMinderWSSDomain that configures the Agent Login Module required to authenticate credentials obtained by Agent authenticators. Configure the SiteMinderWSSDomain by adding a <security-domain-name> element to the standalone.xml file.

**Follow these steps:**

1. Navigate to one of the following locations:
   - **Windows:** *JBOSS_HOME*\standalone\configuration
   - **UNIX:** *JBOSS_HOME*/standalone/configuration
2. Open the standalone.xml file in a text editor.
3. Add *one* of the following <security-domain-name> elements:
   **For JBoss versions before 6.4.5:**

```
<security-domain name="SiteMinderWSSDomain" cache-type="default">
    <authentication>
       <login-module code="com.ca.soa.agent.appserver.jaas.
XMLAgentLoginModule" flag="required"/>
          <module-option name="unauthenticatedIdentity"> value-"anonymous"/>
       </login-module>
    </authentication>
</security-domain>
```

**For JBoss version 6.4.5 or later:**

```
<security-domain name="SiteMinderWSSDomain" cache-type="default">
    <authentication>
       <login-module code="com.ca.soa.agent.appserver.jaas.
XMLAgentLoginModule" flag="required" module="com.ca.siteminder.jbossagent"/>
          <module-option name="unauthenticatedIdentity"> value="anonymous"/>
       </login-module>
    </authentication>
</security-domain>
```

4. Save the file and exit the text editor.

## Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the CA Single Sign-on Agent.

**To restart the JBoss Application Server**

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the CA Single Sign-on Agent.

# OneView Monitor

The following sections describe how to install and configure the the OneView Monitor.

# OneView Monitor Overview

The OneView Monitoring infrastructure consists of a number of modules that enable the monitoring of CA Single Sign-On components. Included is the Monitor process that runs in the context of a Java Runtime Environment (JRE). The Monitor GUI's HTML pages are generated by Java Server Pages (JSPs) and servlets hosted on a ServletExec servlet engine running on the same machine as the Policy Server.

The OneView Monitor utility monitors the following CA Single Sign-On components:

- Policy Server
- Web Agents

# System Requirements for OneView Monitor

The system to which you are configuring the OneView Monitor GUI must meet at least the following system requirements:

- **JDK**—The required version of the Java SDK is installed on the system.
- **Servlet Engine**—The required version of ServletExec is installed on the system.

  **Note:** The Policy Server installation kit includes the installation executable for ServletExec /AS at the following location:

  `thirdparty-tools\servlet-engine-6.0`

- **Web server**—A supported Web server is installed on the system.

For a list of supported CA and third-party components, see the CA Single Sign-On Platform Support Matrix (http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/technical-document-index/ca-siteminder-informational-documentation-index.aspx#/PSM).

# Configure the OneView Monitor

If you did not configure the OneView Monitor GUI when installing the Policy Server, you can configure it using the Policy Server Configuration Wizard.

You can find the following Policy Server Configuration Wizard executables in *siteminder_home* \siteminder\install_config_info for Windows and *siteminder_home*/siteminder/install_config_info for UNIX:

- ca-ps-config.exe

  **Important!** On Windows, if User Account Control (UAC) is enabled, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA Single Sign-On component.

- ca-ps-config.bin

- **siteminder_home**
  Specifies the path to where the Policy Server is installed.

# Limitation of OneView Monitor GUI IIS Web Agent on Same Machine

If the Agent has Registration Services enabled, do not configure the IIS-based OneView Monitor GUI and the IIS Web Agent on the same machine. With this configuration, there is a conflict with the same instance of ServletExec.

# Configure the OneView Monitor on Windows IIS

**Contents**



To configure the OneView Monitor GUI on Windows/IIS complete the following procedures:

1. Read the prerequisites to installing ServletExec on Windows.
2. Install ServletExec/ISAPI on Windows/IIS.

**Note:** The Monitor GUI's HTML pages are generated by Java Server Pages (JSPs) and servlets hosted on a ServletExec servlet engine running on the same machine as the Policy Server.

3. Assign modify permissions to the Internet guest account for the *policy_server_home* \monitor\settings folder.
4. Set permissions for the IIS Users.
5. If you did not have the Policy Server installation program auto-configure the OneView Monitor GUI, configure it by running the Policy Server Configuration Wizard.
6. Start the OneView Monitor service.
7. Access the OneView Monitor GUI.

# Prerequisites to Installing ServletExec on Windows

Consider the following prerequisites before installing ServletExec:

- ServletExec is a third–party product. We recommend that you read the ServletExec documentation before installing the component.

  **Note:** For more information, see the New Atlanta web site (http://www.newatlanta.com).

- The Policy Server requires a 32–bit JDK when installed to a supported 64–bit Windows operating system. If you are installing ServletExec/AS 6.0 to a 64–bit Windows operating system, ServletExec requires a 64–bit JVM. If necessary, install a 64–bit JVM before installing ServletExec.

# Install ServletExec on Windows IIS

**Follow these steps:**

1. If you have a previous version of ServletExec, back it up.
2. Run the ServletExec installation executable.

   **Note:** The Policy Server installation kit includes the installation executable for ServletExec/AS 6.0 at the following location:

   `thirdparty-tools\servlet-engine-6.0`

   **Note:** For more information about upgrading and installing ServletExec, see the New Atlanta documentation (http://www.newatlanta.com).

3. If you installed a 64–bit JVM as a ServletExec/AS prerequisite, complete the following steps:
    a. Use the Services control panel to stop the ServletExec service.
    b. (Optional) Uninstall the 64–bit JVM.
    c. Edit the StartServletExec.bat and StopServletExec.bat files to point to the 32–bit JVM.
    d. Use the Services control panel to start the ServletExec service.
4. Stop and restart the IIS Admin Web service and IIS Web server.

## Set Permissions for IIS Users After Installing ServletExec

Since ServletExec/AS runs as part of the IIS process, it runs as different users at different times. As a result, you must set the following permissions for the ServletExec installation directory and subdirectories.

To set permissions for IIS users after installing ServletExec, Make sure the user that runs IIS (for example, Network Services) has read and write access to the entire directory tree under C:\Program Files\New Atlanta.

# Configure the OneView Monitor on UNIX Sun Java System

**Contents**



To configure the OneView Monitor GUI on a UNIX/Sun Java System complete the following procedures:

1. Read the prerequisites to installing ServletExec.
2. Disable servlets in Sun Java System (Sun One/iPlanet) 6.0.
3. Install ServletExec/AS on UNIX/Sun Java System.
4. If you did not have the Policy Server installation program auto-configure the OneView Monitor GUI, configure it by running the Policy Server Configuration Wizard.
5. Start the OneView Monitor Service.
6. Access the OneView Monitor GUI.

# Prerequisites to Installing ServletExec

CA recommends that you read the ServletExec documentation before installing ServletExec. If ServletExec is not running properly, then the OneView Monitor GUI does not work since it relies on ServletExec's servlet engine.

You can access the ServletExec documentation on the New Atlanta Web site (http://www.newatlanta.com).

# Disable Servlets in Sun Java System 6.0

Ensure you follow the steps in this section before installing ServletExec.

**To disable servlets in Sun Java System 6.0**

1. Open the Sun Java System Enterprise Administration Server home page by entering the following URL in a browser: http://<yourserver.com>:<portnumber>

   - **yourserver.com**
     Specifies the domain name of the Enterprise Administration Server

   - **port**
     Specifies the port number

2. In the Select a Server drop-down menu, select the target server, and then click Manage.

3. Select the Java tab.

4. Deselect Enable Java for class defaultclass and Enable Java Globally and click OK.

5. Stop and restart the Web server so the settings can take effect.

# Install ServletExec AS on UNIX Sun Java System

The Monitor GUI's HTML pages are generated by Java Server Pages (JSPs) and servlets hosted on a ServletExec servlet engine running on the same machine as the Policy Server.

**To install ServletExec**

1. Log in to the UNIX account where you want to install the Policy Server.

   **Note:** You must log in as the same user who installed the Sun Java System Web server.

2. Run the ServletExec AS installer.

**Note:** For more information on running the ServletExec AS installer, refer to New Atlanta Communications' ServletExec documentation. Consider the following before installing ServletExec:

- Make sure you have permission to create a new file in /tmp. New Atlanta recommends installing ServletExec in /usr/local/NewAtlanta. Installing ServletExec in /usr/local /NewAtlanta may change the permissions for the obj.conf file and the Sun Java System start script. After the installation, be sure the owner of obj.conf and the start script is the same user who owns the Web server.
- When prompted, install a Web server adaptor and an instance of ServletExec.
- When prompted, ensure that the installer does not modify the Web server's configuration files. If you let the installer modify the Web server's obj.conf and magnus.conf configuration files, the Web server instance fails to run after you configure the OneView Monitor GUI on this instance.

3. After the installation program completes, restart the Web server.

# Start the OneView Monitor Service

**Follow these steps:**

1. Make sure the IPC port numbers are available.
   The OneView Monitor uses the following port numbers to communicate with the Policy Server processes:
   - Monitoring Agent: 44449
   - Monitor: 44450

   To see which port numbers are unavailable, open a Command Window and enter:

   ```
   netstat -an
   ```

2. Using the Status tab of the Policy Server Management Console, start the Monitor service.

# Access the OneView Monitor GUI

**Follow these steps:**

Enter the following URL in a browser:

http://*server:<portnumber>*/siteminder monitor

- **server**
  Specifies the Web Server's IP Address

- **portnumber**
  Specifies the port number.

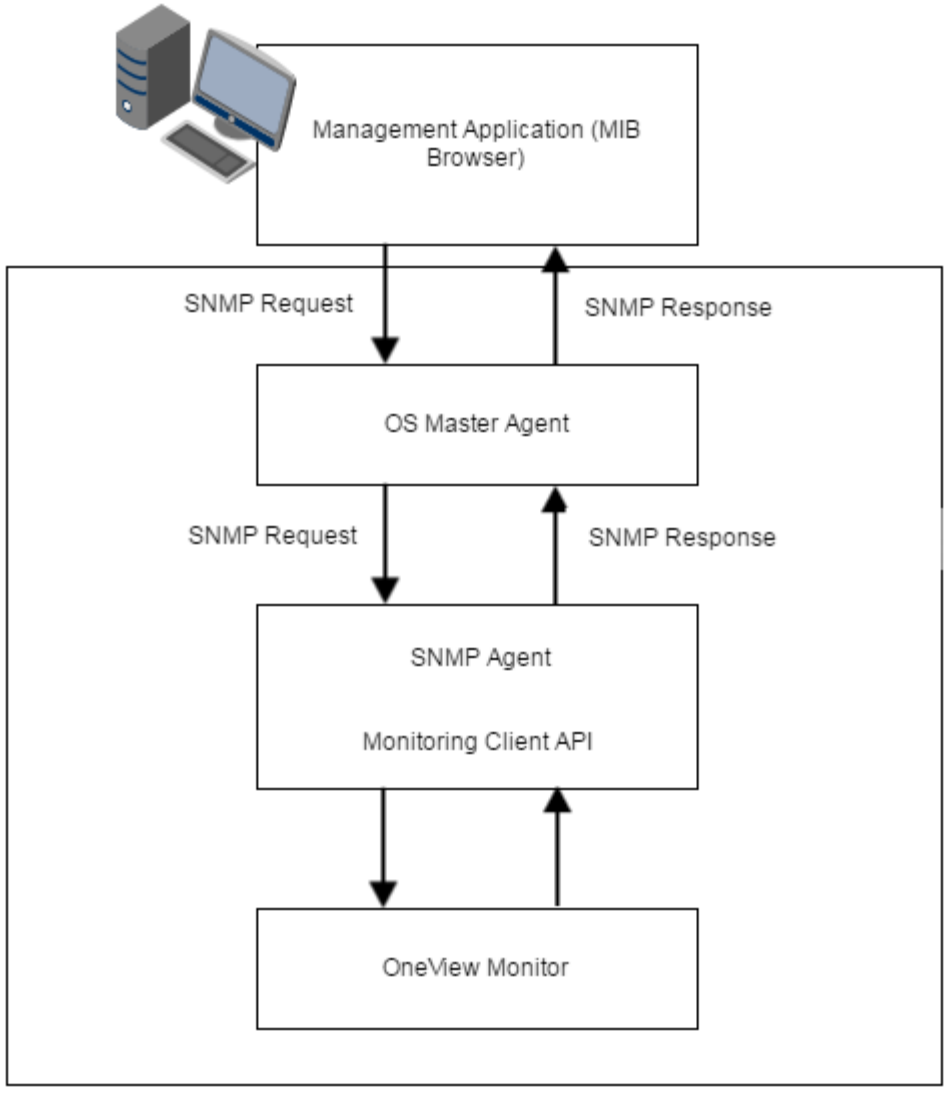
# Monitor a Policy Server Cluster

The OneView Monitor can be configured to monitor a Policy Server cluster when one Policy Server is set up as a centralized monitor for other Policy Servers in a cluster.

# Install and Configure SNMP Support

SNMP support includes a Management Information Base (MIB), an SNMP Agent, and the Event SNMP Trap library. You can configure the SNMP Agent and Event SNMP Trap library independently and enable one or disable the other or vice versa. The SNMP Agent enables monitoring applications to retrieve operational data from the OneView Monitor. The SNMP Agent sends data to the SNMP manager and supports SNMP request handling.

The following figure shows the architecture between the management application, OS Master Agent, SNMP Agent, and the OneView Monitor:

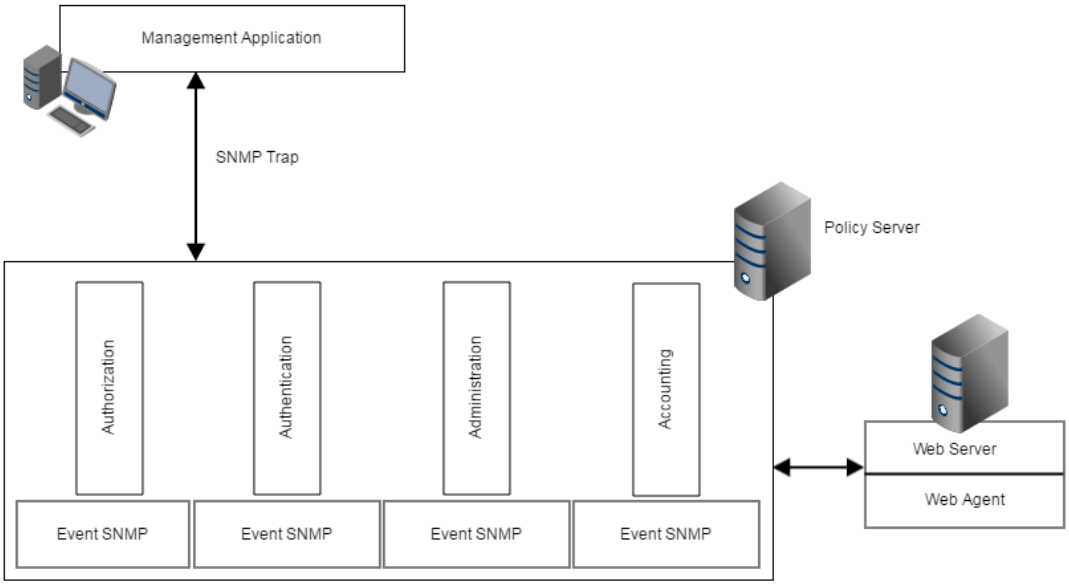



The OS Master Agent, such as the native Solaris SunSolstice Master Agent, invokes the SNMP Agent once you restart the Master Agent. Upon receiving an SNMP request from the management application the OS Master Agent forwards the SNMP request to the SNMP Agent. The SNMP Agent

contacts the OneView Monitor, retrieves the required information using Monitor Client API, and then sends the response to the Master Agent. The Master Agent, in turn, forwards the response to the management application.

If you do not configure the SNMP Agent during the Policy Server installation, all the SNMP files are still installed in case you want to use the Agent later. However, to get the Agent running, you need to manually get the Agent started by configuring the SNMP Agent on a Windows or UNIX system.

The Event SNMP Trap library converts some CA Single Sign-On events into SNMP traps before sending them to the management application as noted in the following figure. The trap library captures events sent by the Policy Server, decides if SNMP traps are to be generated on a given event, and generates a trap.




The following sections describe how to install and configure SNMP support.

# SNMP Prerequisites for Windows and UNIX Systems

**Contents**



You need to have a Master Agent installed with your operating system before installing or using the SNMP Agent.

# Windows Prerequisites

CA Single Sign-on SNMP support on Windows requires the SNMP service. For more information about installing the SNMP service, see the Windows online help system.

# UNIX Systems Prerequisites

The following section details UNIX prerequisites for SNMP support:

## Solaris

You need the native Solaris SunSolstice Master Agent, which comes with the operating system.

## Linux

For the supported Master Agent on Red Hat Advanced Server 3.0, upgrade the net–snmp package to net-snmp-5.1-2.1 or greater.

To upgrade the net–snmp package to net-snmp-5.1-2.1 or greater, use the following setting in the snmpd.conf file for the net–snmpd command:

```
proxy -c public -v 1 localhost:8001 .1.3.6.1.4.1.2552
```

**Note:** After you upgrade the net–snmp package, add proxy support to the snmpd.conf file.

You can find the snmpd.conf file specific to CA Single Sign-On in the following location (The host usually has many snmpd.conf files):

```
/opt/siteminder/etc/snmp/conf/snmpd.conf
```

# Configure the SNMP Agent on Windows

**Follow these steps:**

1. Be sure that the NETE_PS_ROOT environment variable is set to the CA Single Sign-On installation directory. The Policy Server installation program should have already done this.

2. Open *siteminder_home*\config\snmp.conf file and edit the last row to contain the full path to *siteminder_home*\log\snmp.log.

**Note:** You only need to do this if you did not specify the Policy Server installation program to automatically configure SNMP.

**Correct example:** LOG_FILE=C:\Program Files\Netegrity\siteminder\ log\snmp.LOG
**Incorrect example:** LOG_FILE=$NETE_PS_ROOT\log\snmp.log

3. Edit the w*indows_dir*/java_service.ini file.

   **Note:** You only need to do this if you did not specify the Policy Server installation to automatically configure SNMP.

   a. Set SERVICE_BINARY_NAME to the full path name of JavaService.exe.
   **Example:** SERVICE_BINARY_NAME=c:\winnt\JavaService.exe

   b. Set WORKING_DIR to the full path to directory *siteminder_home*\bin:
   **Example:** WORKING_DIR=C:\Program files\Netegrity\siteminder\bin

   c. Set JRE_PATH to the full path of javaw.exe.

4. Run *siteminder_home*\bin\thirdparty\proxyreg.exe to change the registry keys for the apadll.dll and snmp.conf:

   ```
   proxyreg.exe full_path_for_apadll.dll full_path_for_snmp.conf
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   **Example:** proxyreg.exe "c:\program files\netegrity\siteminder\ bin\ thirdparty\apadll.dll" "c:\programfiles\netegrity\ siteminder\ config\ snmp.conf"

5. Run *WINNT dir*/JavaService.exe with the -install option, to register the Netegrity SNMP agent as a WINNT service.

6. Start the Netegrity SNMP agent by using the Windows Services dialog box.

7. Restart the SNMP service.

# How to Configure SNMP Event Trapping on Windows

**Contents**

# Enable SNMP Event Trapping

To enable SNMP event trapping, use the XPSConfig utility to set the event handler library (eventsnmp.dll) to the XPSAudit list. The default location of eventsnmp.dll is *policy_server_home*\bin.

- **policy_server_home**
  Specifies the Policy Server installation location.

After enabling SNMP event trapping, configure the snmptrap.conf file.

# Configure snmptrap.conf

**To configure the SNMP configuration file**

1. Edit snmptrap.conf.

   **Note:** snmptrap.conf is located in *policy_server_home*\config.

   - **policy_server_home**
     Specifies the Policy Server installation location.

2. For the specified trap(s) that you want to receive, uncomment out the appropriate line(s).
3. Specify the IP Address, port number, and community for where you want the trap to be sent.
4. Save the snmptrap.config file with the new changes.
5. Restart the Policy Server.

# Configure the SNMP Agent on UNIX Systems

**Follow these steps:**

1. Ensure the NETE_PS_ROOT environment variable is set to the CA Single Sign-On installation directory. The Policy Server installation program should have already done this.
   **Example:** /home/smuser/siteminder
2. Edit the file /etc/snmp/conf/RunSubagent.sh:
   a. Set the correct JRE path: JAVA_HOME=$INSTALL_HOME/bin/jdk/<required_version>/jre
   b. Set the correct CA Single Sign-On path:
   **Example:** INSTALL_HOME=/home/smuser/siteminder

**Note:** The INSTALL_HOME variable should contain the full path for the CA Single Sign-On installation directory.

3. Restart the SNMP daemon on Solaris

   a. Become root.

   b. Goto /etc/rc3.d.

   c. Execute the S76snmpdx script twice, as follows:
   sh S76snmpdx stop to stop the running Solaris master agent.
   sh S76snmpdx start to start the Solaris master agent and Netegrity subagent.

# How to Configure SNMP Event Trapping on UNIX Systems

**Contents**



## Enable SNMP event trapping

To enable SNMP event trapping, use the XPSConfig utility to set the event handler library (libeventsnmp.so) to the XPSAudit list. The default location of libeventsnmp.so is *policy_server_home* /lib.

- **policy_server_home**
  Specifies the Policy Server installation location.

After enabling SNMP event trapping, configure the snmptrap.config file.

## Configure snmptrap.config

**To configure snmptrap.config**

1. Edit snmptrap.config, which is located in /home/smuser/siteminder/config.
2. For the specified trap(s) that you want to receive, uncomment out the appropriate line(s).
3. Specify the IP Address, port number, and community for where you want the trap to be sent.
4. Save the snmptrap.config file with the new changes.

5. Restart the Policy Server.

# Test SNMP Gets for Red Hat Enterprise Linux Advanced Server

You should test SNMP Gets after configuring SNMP.

**To test SNMP Gets**

1. Start the native SNMP master agent. On Red Hat AS, the master agent is not started automatically on start up as is the case on Solaris and HP-UX. To start the master agent, go to the NETE_PS_ROOT/etc/snmp/conf/ directory and run the following command. Run the command as root:

   ```
   K50snmpd start
   ```

2. Start the Netegrity subagent using the following command (run as root):

   ```
   sh /etc/init.d/NetegrityAgent
   ```

3. To stop the Netegrity subagent on Red Hat AS, run the following command as root:

   ```
   sh $NETE_PS_ROOT/etc/snmp/conf/StopSubagent.sh
   ```

# Unattended Installations

After you install a component the first time, you can install the component on other systems using an *unattended installation*. An unattended installation lets you complete the installation without user intervention. You can use an unattended installation with the following components:

- Policy Server
- Administrative UI
- Report Server

After you install a component the first time, a properties file for an unattended installation is also installed. Each component is associated with its own properties file or files. The following guidelines apply to all properties files. Review them before starting an unattended installation:

- Back up the default properties file before modifying it.
- Do not add extra spaces between a parameter name, the equal sign (=), and the parameter value.
- Save the file after you change it.
- Do not manually edit encrypted passwords. These passwords are encrypted for security reasons and cannot be edited in plain text. If you want to add plain text passwords, comment out the encrypted password parameter and uncomment the plain text reference.

The default parameters in the file reflect the information that was entered during the initial installation. Use a text editor to modify the parameters in a default properties file.


- Policy Server Properties File (see page 1014)
- Administrative UI Properties Files (see page 1014)
- Reporting Properties File (see page 1015)


# Policy Server Properties File

The Policy Server properties file has the following default name and location:

**Name:** ca-ps-installer.properties

**Location:** *policy_server_home*\siteminder\install_config_info
*policy_server_home* specifies the Policy Server installation path.

# Administrative UI Properties Files

An unattended Administrative UI installation requires only one properties file. The required file depends on the installation option you select. The properties files are:

- **prerequisite-installer.properties**
  If you are installing using the stand-alone installation, use this file. This file is only available if the Administrative UI was previously installed using the stand-alone installation option.

  Location: *admin_ui_home*\siteminder\adminui\install_config_info
  *admin_ui_home* specifies the Administrative UI installation path.

- **smwamui-installer.properties**
  If you are installing the UI to an external application server, use this file.

  Location: *admin_ui_home*\siteminder\adminui\install_config_info
  *admin_ui_home* specifies the Administrative UI installation path.

# Reporting Properties File

An unattended report installation requires a response file for the Report Server and a response file for the Report Server Configuration Wizard. These files have the following names and default locations:

- **cabiresponse.ini**
  Use this file to install the Report Server.

  Location: *report_server_home*\CA\SC\CommonReporting3

- **reportserver_config_installer.properties**
  Use this file to install the reporting templates.

  Location: *report_server_home*\CommonReporting3\install_config_info

*report_server_home* specifies the Report Server installation path.

For unattended installation instructions, refer to the section for the component you want to install.

# Policy Server Unattended Installation

**Contents**

- Stop an Unattended Policy Server Installation (see page 1025)

To run an unattended Policy Server install, complete the following procedures:

1. Review the unattended installation guidelines.
2. Copy the Policy Server properties file from the Policy Server host system.
3. Complete one of the following steps:
   - If you are reinstalling the Policy Server, copy the file to a temporary location.
   - If you are installing the Policy Server to a new system, copy the file to a temporary location on that system.

   **Note:** (UNIX) Be sure that the UNIX user has the appropriate permissions to install from this directory.
4. Copy the Policy Server installation media to the same location as the properties file.
5. Modify the Policy Server installer properties file.
6. Run the Policy Server installer.
7. Verify the Policy Server installation.

# Modify the Policy Server Installer Properties Files

You modify the Policy Server installer properties file to define installation variables. The default parameters, passwords, and paths in this file reflect the information you entered during the initial Policy Server installation.

**Important!** The properties template includes a variable that specifies the Policy Server's FIPS mode of operation: CA_SM_PS_FIPS140. If you are reinstalling the Policy Server, do not modify the value of the variable. If required, change the FIPS mode of operation after reinstalling the Policy Server. More information on changing the Policy Server's FIPS mode of operation exists in the *Upgrade Guide*.

## General Policy Server Information

The General Information section allows you to set the following:

- **DEFAULT_INSTALL_DIR**
  Specifies the location of the Policy Server installation.

- **DEFAULT_SHORTCUTS_DIR**
  Specifies the location of the CA Single Sign-On program icon. The icon feature only works on Windows.
  **Example:** C:\\Documents and Settings\\All Users\\Start or /CA Single Sign-On

- **DEFAULT_JRE_ROOT**
  Specifies the JRE installation location.

- **DEFAULT_BROWSER**
  (UNIX only) Specifies the installation location of the browser.
  **Example:** /usr/dt/appconfig/netscape/netscape

- **DEFAULT_SMPROFILE_CHOICE**
  (UNIX only) Specifies if smprofile.ksh should be added to the .profile file. Specify **true** for yes; specify **false** for no.

- **DEFAULT_ENCRYPTKEY**
  Allows you to enter a cleartext encryption key, which secures data sent between the Policy Server and the policy store.

  **Note:** If you comment out the ENCRYPTED_ENCRYPTKEY parameter and uncomment DEFAULT_ENCRYPTKEY, then the unattended installer uses the cleartext encrypt key value from DEFAULT_ENCRYPTKEY. The DEFAULT_ENCRYPTKEY parameter is commented out by default after the initial Policy Server installation.

- **ENCRYPTED_ENCRYPTKEY**
  Shows the encrypted encryption key, which secures data sent between the Policy Server and the policy store. You entered this key during the initial Policy Server installation and cannot change it.

  **Important!** Do not modify this encrypted value since any change will break the communication between the Policy Server and policy store when you run an unattended installation.

  If you comment out the DEFAULT_ENCRYPTKEY parameter and uncomment ENCRYPTED_ENCRYPTKEY, then the unattended installer uses the encrypted encryption key value from ENCRYPTED_ENCRYPTKEY.

- **CA_SM_PS_FIPS140**
  Specifies the Policy Server's FIPS mode of operation.
  **Values:** COMPAT, MIGRATE, or ONLY

  **Important!** Do not modify the value if you are reinstalling the Policy Server.

## Policy Server Features

The Feature Selection section lets you set the following parameters:

- **DEFAULT_OVMGUI_CHOICE**
  Determines if the Policy Server installer configures the OneView Monitor GUI on the selected web server.
  **Values:**
  - **true**
    The installer configures the OneView Monitor GUI.
    Setting this value to true requires you to configure additional settings under OneView Monitor GUI and Web Servers.
  - **false**
    The installer does not configure the OneView Monitor GUI.
- **DEFAULT_WEBSERVERS_CHOICE**
  Determines if the Policy Server installer configures the Federation Security Services UI with a specified web server.
  **Values:**
  - **true**
    The installer configures the component with the specified web server.
    Setting this value to true requires you to configure additional settings under Web Servers.
  - **false**
    The installer does not configure the component with a web server.
- **DEFAULT_SNMP_CHOICE**
  Determines if the Policy Server installer configures SNMP support with the Policy Server.
  **Values:**
  - **true**
    The installer configures SNMP support.
    Setting this value to true requires you to configure additional settings under SNMP.
  - **false**
    The installer does not configure SNMP support.
- **DEFAULT_POLICYSTORE_CHOICE**
  Determines if the Policy Server installer configures a policy store automatically.
  **Values:**
  - **true**
    The installer configures a policy store.
    Setting this value to true requires you to configure additional settings under Policy Store.
  - **false**
    The installer does not configure a policy store.

## OneView Monitor

If you set the DEFAULT_OVMGUI_CHOICE parameter to true, then set the following:

- **DEFAULT_JDK_ROOT**
  Specifies the JDK installation location.

- **DEFAULT_SERVLETEXEC_INSTANCE_NAME**
  (UNIX only) Specifies the name of the ServletExec instance.
  **Example:** se-testmachine-60psGUI

- **DEFAULT_SERVLETEXEC_ROOT**
  Specifies the ServletExec installation location.
  **Example:** C:\\Program Files\\New Atlanta\\ServletExec ISAPI or /export/NewAtlanta /ServletExecAS

- **DEFAULT_SERVLETEXEC_PORT**
  (UNIX only) Specifies the port number of the ServletExec instance.
  **Example:** 7676

## SNMP

If you want to modify the SNMP password, do the following:

- **DEFAULT_ROOT_PW**
  Allows you to enter a cleartext SNMP password for the UNIX system's root user. If you comment out the ENCRYPTED_ROOT_PW parameter and uncomment DEFAULT_ROOT_PW, then the unattended installer uses the cleartext SNMP password from DEFAULT_ROOT_PW.
  Default: The DEFAULT_ROOT_PW parameter is commented out after the initial Policy Server installation.

- **ENCRYPTED_ROOT_PW**
  Shows the encrypted SNMP password for the UNIX system's root user. You entered this password during the initial UNIX Policy Server installation and cannot change it.

  **Important!** Do not modify this encrypted password since any change will break the communication between the Policy Server and the SNMP Agent. If you comment out the DEFAULT_ROOT_PW parameter and uncomment ENCRYPTED_ROOT_PW, then the unattended installer uses the encrypted password from ENCRYPTED_ROOT_PW.

## Policy Store

If you set the DEFAULT_POLICYSTORE_CHOICE parameter to true, then set the following parameters:

- **DEFAULT_POLICYSTORE_TYPE**
  Specifies the type of store that is to function as the policy store.
  **Values:**

- **LDAP**
  Specifies an LDAP policy store.

- **RDB**
  Specifies an ODBC policy store.

- **DEFAULT_POLICYSTORE_IP**
  (LDAP) Specifies the IP address or name of the LDAP directory server host system.
  **Example:** 172.16.0.0

- **DEFAULT_POLICYSTORE_PORT**
  (LDAP) Specifies the port on which the LDAP directory server is listening.
  **Example:** 1356.

- **DEFAULT_POLICYSTORE_ADMINDN**
  (LDAP) Specifies the LDAP user name of an administrator who has permission to:

  - Create schema.

  - Create, read, modify, and delete objects in the LDAP tree under the policy store root object.

  **Example:** cn=Directory Manager.

- **DEFAULT_POLICYSTORE_ADMINPW**
  (LDAP) Lets you enter a cleartext password for the administrator of the LDAP directory server. If you comment ENCRYPTED_POLICYSTORE_ADMINPW and uncomment DEFAULT_POLICYSTORE_ADMINPW, then the unattended installer uses the cleartext password from DEFAULT_POLICYSTORE_ADMINPW.
  **Default:** This parameter is commented out after the initial Policy Server installation.

- **ENCRYPTED_POLICYSTORE_ADMINPW**
  (LDAP) Represents the encrypted password for the administrator of the LDAP directory server. This password was entered the last time the Policy Server installer configured the policy store. You can use the existing encrypted value to provide the LDAP administrator password for the new policy store. This password cannot be changed.

  **Important!** Do not modify this password. The password is encrypted. If you comment out the DEFAULT_POLICYSTORE_ADMINPW and uncomment ENCRYPTED_POLICYSTORE_ADMINPW, then the installer uses the encrypted password from ENCRYPTED_POLICYSTORE_ADMINPW.

- **DEFAULT_POLICYSTORE_ROOTDN**
  (LDAP) Specifies the root DN of the LDAP directory server.
  **Example:** o=example.com.

- **DEFAULT_POLICYSTORE_USER_CHOICE**
  (LDAP) The DEFAULT_POLICYSTORE_ADMINDN parameter requires an LDAP administrator user name that has permission to create the schema. By default, the Policy Server uses this account to manage the policy store. An alternate LDAP user account can manage data in the policy store

after the policy store is configured. The alternate account must have permission to create, read, modify, and delete objects.
**Values:**

- **true**
  Specifies that an alternate LDAP user account is to manage the policy store after the policy store is configured.

- **false**
  Specifies that the LDAP administrator user account, which the DEFAULT_POLICYSTORE_ADMINDN parameter specifies, is to manage the policy store after the policy store is configured.

- **DEFAULT_POLICYSTORE_USERDN**
  (LDAP) Specifies the DN of the alternate LDAP user account.
  **Example:** uid=SMAdmin,ou=people,o=security.com.

- **DEFAULT_POLICYSTORE_USERPW**
  (LDAP) Lets you enter a cleartext password for the alternate LDAP user. If you comment ENCRYPTED_POLICYSTORE_USERPW and uncomment DEFAULT_POLICYSTORE_USERPW, then the unattended installer uses the cleartext password from DEFAULT_POLICYSTORE_USERPW.
  **Default:** The DEFAULT_POLICYSTORE_USERPW parameter is commented out after the initial Policy Server installation.

- **ENCRYPTED_POLICYSTORE_USERPW**
  (LDAP) Represents the encrypted password for the alternate LDAP user. This password was entered the last time the Policy Server installer configured the policy store. You can use the existing encrypted value to set the alternate administrator password for the new policy store. This password cannot be changed.

  **Important!** Do not modify this password. This password is encrypted.

  If you comment DEFAULT_POLICYSTORE_USERPW and uncomment ENCRYPTED_POLICYSTORE_USERPW, then the installer uses the encrypted password from ENCRYPTED_POLICYSTORE_USERPW.

- **DEFAULT_INIT_POLICYSTORE_CHOICE**
  (LDAP/RDB) Specifies if the Policy Server installer must initialize the policy store.
  **Values:**

  - **true**
    The installer initializes the policy store.

  - **false**
    The installer does not initialize the policy store.

- **DEFAULT_SM_ADMINPW**
  (LDAP/RDB) Lets you enter a cleartext password for the super user account.
  If you comment ENCRYPTED_SM_ADMINPW and uncomment DEFAULT_SM_ADMINPW, then the installer uses the cleartext password from DEFAULT_SM_ADMINPW.
  **Default:** The DEFAULT_SM_ADMINPW parameter is commented out after the initial Policy Server installation.

- **ENCRYPTED_SM_ADMINPW**
  (LDAP/RDB) Represents the encrypted password for the super user account. This password was entered the last time the Policy Server installer configured the policy store. You can use the existing encrypted value to set the super user password for the new policy store. This password cannot be changed.

  Do not modify this password. It is encrypted.

  If you comment DEFAULT_SM_ADMINPW and uncomment ENCRYPTED_SM_ADMINPW, then the installer uses the encrypted password from ENCRYPTED_SM_ADMINPW.

- **DEFAULT_RDB_DSN**
  (RDB) Specifies the name of the DSN that the Policy Server installer creates.

- **DEFAULT_RDB_DBSERVER**
  (RDB) Specifies the IP address or name of the database host system.

- **DEFAULT_RDB_DBNAME**
  (RDB) Specifies one of the following values:

  - (SQL Server) the named instance of the SQL Server that is to function as the policy store.
  - (PostgreSQL) the named instance of the PostgreSQL Server that is to function as the policy store.
  - (Oracle) the service name of the Oracle database that is to function as the policy store.

- **DEFAULT_RDB_PORT**
  (RDB) Specifies the port on which the database is listening.

- **DEFAULT_RDB_USER_NAME**
  (RDB) Specifies the name of the database administrator account that has permission to:

  - Create schema
  - Create, read, modify, and delete objects.

- **DEFAULT_RDB_DBTYPE**
  Specifies the type of database that is to function as the policy store.**Values:**

  - **DB_MSSQL**
    Specifies a SQL Server policy store.
  - **DB_ORACLE**
    Specifies an Oracle policy store.

- **DB_POSTGRE**
  Specifies a PostgreSQL policy store.

- **DEFAULT_RDB_PASSWORD**
  (RDB) Lets you enter a cleartext password for the database administrator.
  **Default:** This parameter is commented out after the initial Policy Server installation.

  If you comment ENCRYPTED_RDB_PASSWORD and uncomment DEFAULT_RDB_PASSWORD, then the installer uses the cleartext password from DEFAULT_RDB_PASSWORD.

- **ENCRYPTED_RDB_PASSWORD**
  (RDB) Represents the encrypted value of the database administrator password that was entered the last time that the installer configured the policy store.**Default:** This parameter is uncommented. The installer uses this value, unless you comment this parameter and uncomment DEFAULT_RDB_PASSWORD.

- **DEFAULT_KEYSTORE_CONFIG**
  Specifies if the installer must collocate the key store with the policy store.
  **Values:**

  - **true**
    The installer collocates the key store with the policy store.

  - **false**
    The installer does not configure a key store. You configure a stand–alone key store after configuring the policy store.

- **DEFAULT_SMKEYDB_IMPORT_CHOICE**
  Specifies if the default CA certificates must be imported into the certificate data store.
  **Values:**

  - **true**
    Import the default CA certificates.

  - **false**
    Do not import the default CA certificates.

## Enhanced Session Assurance with DeviceDNA™ Settings

The following items apply to Enhanced Session Assurance with DeviceDNA™:

**MASTER_KEY=**
Specifies the master encryption key for the advanced authentication server (which runs on the CA Access Gateway). Stores the master encryption key in plain-text format.

**ENCRYPTED_MASTER_KEY=**
Specifies the master encryption key for the advanced authentication server (which runs on the CA Access Gateway). Stores the master encryption key in an encrypted format.

**IS_SA_ENABLED=true**
Indicates if Enhanced Session Assurance with DeviceDNA™ is enabled. Do *not* edit this item.

# Run the Policy Server Installer

Run an unattended installation to install the Policy Server without user intervention. Install the Policy Server using the installation media on the Technical Support site.

## Windows

Run the following command from the directory where you copied the Policy Server installation executable and the properties file:

*installation_media* -f ca-ps-installer.properties -i silent

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

- *installation_media* specifies the Policy Server installation executable.

**Note:** If the properties file is not in the same directory as the installation media, specify its location. Use double quotes if the argument contains spaces.

- **-i silent s**pecifies that the installer run silently.

**Example:**

ca-ps-*version*-win32.exe -f "C:\Program Files\CA\siteminder\install_config_info\ca-ps-installer.properties" -i silent

The installation begins. The installer uses the parameters that you specified in the properties file to install the Policy Server.

## UNIX

**Follow these steps:**

1. Open a shell.
2. Run the following command from the directory to which you copied the Policy Server installation executable and Policy Server installation properties file:
   ./installation_media -f ca-ps-installer.properties -i silent

   *installation_media* specifies the Policy Server installation executable.

   **-i silent s**pecifies that the installer run silently.

The installation begins. The installer uses the parameters that you specified in the properties file to install the Policy Server.

# Stop an Unattended Policy Server Installation

You stop an unattended Policy Server installation to prevent the Policy Server from installing on the specified Windows system.

To stop the installation:

**Windows**
Use the Windows Task Manager to stop the following processes:

ca-ps-*version*-win32.exeps_install.exe

**UNIX**
Press Ctrl+C.

# Administrative UI Unattended Installation

**Contents**



After running an attended installation, you can run an unattended Administrative UI installation. The attended installation puts down properties files used by the unattended installation.

The general process for running an unattended UI installation is:

1. Determine whether you are running a stand-alone unattended installation or an installation for external application servers.
2. Modify the properties file.
3. Run the installer.
4. Register the Administrative UI with a Policy Server.

# Installation Media Requirements

Consider the following items before running an unattended installation of the Administrative UI:

- Use the installation media from the CA Support site (http://support.ca.com/).

- The installation zip contains a layout.properties file at the same level as the installation media. If you move the installation media after extracting the installation zip, move the layout.properties file to the same location or the installation fails.

- The installation zip contains the prerequisite installer and the Administrative UI installer. If you are installing the Administrative UI using the prerequisite installer, place both executables in the same location.

# Run an Unattended Standalone UI Installation

To run an unattended standalone UI installation, modify the **prerequisite-installer.properties** file to define installation variables. This file is only available if the Administrative UI was previously installed using the stand-alone installation option. The default values reflect the information entered during the last attended UI installation.

## Modify the Prerequisite Installer Properties File

The prerequisite installer properties file is for standalone installations only. To modify this file:

1. Go to the directory *admin_ui_home*\siteminder\adminui\install_config_info
   *admin_ui_home* specifies the Administrative UI installation path.

2. Open the file in a text editor and modify the settings.

**General Information**
Specify the root folder of the Administrative UI installation.

DEFAULT_INSTALL_FOLDER
Specifies the root folder under which all sub-folders are created during the installation.

**Server Port Information**
This section identifies the HTTP server port.

DEFAULT_APP_SERVER_PORT
Specifies the port on which JBoss listens for HTTP requests.

**SSL Port Information**
This section identifies HTTPS server port.

DEFAULT_APP_SERVER_SSL PORT
Specifies the port on which JBoss listens for HTTPS requests.

**Server Name**
Names the server hosting the Administrative UI. This is the server on which you run the unattended installation, not the server where you ran an attended installation.

DEFAULT_APP_SERVER_HOST
Specifies the fully qualified name of the Administrative UI host system.

**End User License Agreement**
The prerequisite installer requires you to accept a Lesser General Public License. The JBoss License Agreement (JBossORG-EULA.txt) is available on the Administrative UI host system where you ran an *attended* installation. Navigate to the following location and review the license:

*admin_ui_home*\siteminder\adminui.
*admin_ui_home* specifies the Administrative UI installation path.

ACCEPT_LGPL_EULA
Indicates if you accept the license agreement. Accept the License Agreement by changing the value of ACCEPT_LGPL_EULA to YES.

## Run an Unattended Standalone Installation (Windows and UNIX)

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Follow these steps:**

1. Determine which properties file to use:
   - **Windows**: Only the prerequisite-installer.properties file. The prerequisite installer kicks off the Administrative UI installer.
   - **UNIX**: The prerequisite-installer.properties and the smwamui-installer.properties
2. Navigate to the directory *admin_ui_home*\siteminder\adminui\install_config_info on the Administrative UI system where you ran the attended installation.
3. Copy the properties file(s) from the Administrative UI host system to a temporary location on the new host system.
4. Modify the properties file(s). Refer to the parameter descriptions in the previous section.
   In the prerequisite-installer.properties file, verify that the ACCEPT_LGPL_EULA parameter is set to YES
5. Go to the location where you extracted the installation zip file for the attended installation. Copy the following files to the same location where you copied the properties file(s):
   - installation executables (prerequisite and UI)
   - layout.properties file
6. From the directory where you copied all the files, run the prerequisite installer by entering the following command:

   ```
   prerequisite_installation_media -f properties_file -i silent
   ```

**Example:** adminui-pre-req-12.52-sp02-win32.exe -f "C:Program Files\CA\siteminder\adminui\install_config_info\prerequisite-installer.properties" -i silent

For Windows systems, the prerequisite installer automatically launches the Administrative UI installer

7. (UNIX only): Run the Administrative UI installation by entering the following command. On UNIX systems, the prerequisite installer does not automatically launch the Administrative UI installer.

   ```
   installation_media -f properties_file -i silent
   ```

   **Example:** ca-adminui-12.52-sp02-win32.exe -f "C:Program Files\CA\SiteMinder\adminui\install_config_info\smwamui-installer.properties" -i silent

   *installation_media s*pecifies the name of the Administrative UI installation executable used to first install the UI.
   **-f** *properties_file* Specifies the path to the properties file. The path must include the properties file name. The Administrative UI installation option determines the properties file that you use. Use double quotes if the argument contains spaces.
   **-i silent s**pecifies that the installer run silently.

8. Register the Administrative UI with a Policy Server. Refer to the Administrative UI installation instructions for registration procedures.

# Run an Unattended Installation on External Application Servers

To run an unattended installation on an external application servers, use the smwamui-installer. properties file.

## Modify the Administrative UI Installer Properties File

If you are installing the Administrative UI to an existing application server infrastructure, modify the **smwamui-installer.properties** file. The default values in this file reflect the information entered during the last Administrative UI installation.

To modify this file:

1. Go to the directory *admin_ui_home*\siteminder\adminui\install_config_info
   *admin_ui_home s*pecifies the Administrative UI installation path.

2. Open the file in a text editor and modify the settings.

**General Information**
Specify the root folder of the Administrative UI installation:

DEFAULT_INSTALL_FOLDER
Specifies the root folder under which all sub-folders are created during the installation.

**Application Server Information**
This section lets you specify information about the application server to which the Administrative UI is to be deployed. This section contains the following parameters:

DEFAULT_APP_SERVER
Specifies the application server type. This parameter uses the following settings:

- **JBoss**
  Specifies that the application server type is JBoss.
- **WebLogic**
  Specifies that the application server type is WebLogic.
- **WebLogic9**
  Specifies that the application server type is WebLogic9.
- **WebSphere**
  Specifies that the application server type is WebSphere.

DEFAULT_NETE_JAVA_HOME
Specifies the path to the required JDK or JRE for the application server. This value depends on the type of application server:

- **JBoss**
  Specifies the path to the minimum version of the required JDK.
- **WebLogic**
  Specifies the path to the minimum version of the required JDK or JRE.
- **WebSphere**
  Specifies the path to the minimum version of the required JDK or JRE.

DEFAULT_APP_SERVER_URL
Specifies the fully qualified URL of the system on which the application server is installed.

**JBoss Information**
This section has the following parameters for JBoss.

**DEFAULT_JBOSS_FOLDER**
Specifies the path to the JBoss installation directory. If you did not enter JBoss as the value of DEFAULT_APP_SERVER, a value is not required for this section.

**WebLogic Information**
This section has the following parameters for WebLogic. If you do not enter WebLogic9 as the value of DEFAULT_APP_SERVER, values are not required for this section.

DEFAULT_BINARY_FOLDER
Specifies the path to the WebLogic installation directory.

DEFAULT_DOMAIN_FOLDER
Specifies the path to the WebLogic domain you created for the Administrative UI.

DEFAULT_SERVER_NAME
Specifies the name of the WebLogic server on which the WebLogic domain is configured.

**WebSphere Information**
This section lets you specify additional information about WebSphere. If you do not enter WebSphere6 as the value of DEFAULT_APP_SERVER, values are not required for this section.

This section has the following parameters:

DEFAULT_WEBSPHERE_FOLDER
Specifies the path to the WebSphere installation directory.

DEFAULT_WAS_NODE
Specifies the name of the node in which the application server is located.

DEFAULT_WAS_SERVER
Specifies the name of the application server.

DEFAULT_WAS_CELL
Specifies the name of the cell in which the application server is located.

WAS_PROFILE
Specifies the name of the profile being used for the Administrative UI.

## Run an Unattended Installation on External Application Servers (Windows and UNIX)

**Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

**Follow these steps:**

1. Navigate to the directory *admin_ui_home*\siteminder\adminui\install_config_info on the Administrative UI system where you ran the attended installation.
2. Copy the smwamui-installer.properties file from the Administrative UI system to a temporary location on the new host system.
3. Modify the smwamui-installer.properties file. Refer to the parameter descriptions in the previous section.
4. Go to the location where you extracted the installation zip file for the attended installation. Copy the following files to the same location where you copied the properties file(s):
   - UI installation executable
   - layout.properties file
5. From the directory where you copied all the files, run the following command:

```
installation_media -f properties_file -i silent
```

**Example:** ca-adminui-12.52-sp02-win32.exe -f "C:Program Files\CA\SiteMinder\adminui\install_config_info\smwamui-installer.properties" -i silent

*installation_media* specifies the name of the Administrative UI installation executable used to first install the UI.
**-f** *properties_file* specifies the path to the properties file. The path must include the properties file name. The Administrative UI installation option determines the properties file that you use.
**-i silent** specifies that the installer run silently.

6. Register the Administrative UI with a Policy Server. Refer to the Administrative UI installation instructions for registration procedures.

# Report Server Unattended Install

**Contents**



To run an unattended Report Server installation, complete the following procedures:

1. Review the installation checklists (see page 491).
2. Gather the required information for the installer.
3. Review the guidelines for silent installation.
4. Locate the Report Server response file and Report Server Configuration Wizard properties file.
5. Copy the response and properties files to a temporary location on the Report Server host system.
6. Copy the Report Server and the Report Server Configuration Wizard installation media to a temporary location on the Report Server host system.

   **Note:** You can download the installation media from the Technical Support site.

7. Modify the Report Server response file for Windows (see page 1032)or UNIX (see page 1044).
8. Modify the Report Server Configuration Wizard properties file (see page 1053).

9. Run the Report Server installer (see page 1054).
10. Run the Report Server Configuration Wizard.
11. Register the Report Server.

# Modify the Report Server Response File for Windows

You modify the installer response file to define installation variables. The default values reflect the information that was entered when a Report Server was installed and a response file was saved.

**Important!** Consider the following items before you modify the file:

- Some of the parameters identify the type of database that is to function as the report (cms) database. Do not use the properties file to change the type of database that the installer is to configure. To change the report database type, use the installer to create another properties file.
- Passing a parameter in the command line overrides the respective setting in the response file.
- The installer generates some parameters automatically. Do not modify the following parameters:
  - DATABASECONNECT
  - DATACONNECTION
  - DATABASE_AUDIT_CONNSVR
  - INSTALLDBTYPE
  - INSTALL_DB_TYPE
  - INSTALLLEVEL
  - INSTALLSWITCH
  - NEWCMSPASSWORD
  - PRIVILEGED
  - SINGLESERVER
  - SKIP_DEPLOYMENT
  - WCADOTNETINSTALL
  - WCAJAVAINSTALL

- WCATOMCATINSTALL
- WDEPLOY_LANGUAGES
- WDEPLOY_LATER
- WEBSITE_METABASE_NUMBER
- WEBSITE_PORT
- ADDSOURCE
- ADVERTISE

## Install

This section details the parameters in the Install section of the response file.

- **AS_ADMIN_IS_SECURE**
  Specifies that an administrator credential must be passed to access the web application server.
  **Recommended setting:** Leave the value empty. This parameter applies to a web application server that CA Single Sign-On does not support.

- **AS_ADMIN_PASSWORD**
  Specifies the password of the administrator account that accesses the web application server. If the embedded version of Apache Tomcat is installed, the value of this parameter is empty. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** Leave the value empty.

- **AS_ADMIN_PORT**
  Specifies the port on which the web application server is listening. If the embedded version of Apache Tomcat is installed, the value defaults to 8080. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** 8080
  Be sure that this value matches the value of TOMCAT_CONNECTION_PORT.

- **AS_ADMIN_USERNAME**
  Specifies the account name of the administrator account that is used to access the web application server. If the embedded version of Apache Tomcat is installed, the value defaults to the following value:

  ```
  admin
  ```

  CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** admin

- **AS_DIR**
  Specifies the path to which the embedded version of Apache Tomcat is installed. The path is automatically set using the installation directory.

  
  **Note:** Be sure that you escape the backslashes in the path with a backslash character.

**Example:** "C:\\Program Files\\CA\\SC\\CommonReporting3\\Tomcat55"

- **AS_INSTANCE**
  Specifies the name of the web application server instance. If the embedded version of Apache Tomcat is installed, the value defaults to localhost. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** localhost

- **AS_SERVER**
  Specifies the type of Java web application server to deploy. If the embedded version of Apache Tomcat is installed, the value defaults to tomcat55. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** tomcat55

- **AS_SERVICE_NAME**
  If the web application server is installed as a service on Windows, specifies the name of the service. If the embedded version of Apache Tomcat is installed, the value defaults to the following value:

  ```
  BOE120Tomcat
  ```

  CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** BOE120Tomcat

- **AS_VIRTUAL_HOST**
  If you are deploying the Report Server to a virtualized environment, specifies that the virtual host to which the application must be bound.
  **Recommended setting:** Leave the value empty.

- **AUDITINGALLOWED**
  If you are configuring a Microsoft SQL Server or Oracle report database, specifies that the Content Management Server Auditing Database component can be configured.

  - **0**
    Indicates that auditing is allowed.

  - **1**
    Indicates that auditing is not allowed.

  **Recommended setting:** 0

- **AUDITINGENABLED**
  If you are configuring a Microsoft SQL Server or Oracle report database, specifies that the Content Management Server Auditing Database is enabled. This parameter is not related to CA Single Sign-On audit–based reports.
  The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On. A CA Single Sign-On audit database is required to run audit–based reports.

  - **0**
    Indicates that auditing is enabled.

  - **1**
    Indicates that auditing is not enabled.

**Recommended setting:** 1

- **CANODE**
  Specifies the name of the Service Intelligence Agent (SIA) node.

  
  **Note:** Do not use spaces or non–alphanumeric characters.

- **CADPORT**
  Specifies the port to which the SIA must connect and listen for requests.
  **Default:** 6410

- **CLIENTAUDITINGREPORT**
  If the Content Management Server Auditing Database component is enabled, specifies the port on which the auditing service must listen. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:** Leave the default value (6420).

- **CLIENTLANGUAGE**
  Specifies the language pack to install.
  **Recommended setting:** EN

- **CLUSTERCMS**
  Specifies if you are adding servers to an existing Content Management Server. If you must change this value, we recommend using the installer to create another response file.
  **Recommended setting:** False

- **CMSPASSWORD**
  Specifies the password for an existing SAP BusinessObjects Enterprise administrator account. This parameter only applies to a custom or web tier installation.
  **Recommended setting:** Leave the value empty.

- **DATABASEAUDITDRIVER**
  Specifies which driver to use for the Content Management Server Auditing Database. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended settings:**

  - If you are installing the embedded version of MySQL, leave the default value (MySQLDatabaseSubSystem).

  - If you are configuring and a Microsoft SQL Server or Oracle report database, leave the value empty.

- **DATABASEAUTHENTICATION**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

- **DATABASEAUTHENTICATION_AUDIT**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

- **DATABASECONNECT**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

- **DATABASECONNECT_AUDIT**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

- **DATABASEDB**
  Specifies the name of the report database.
  **Recommended setting:**

  - If you are installing the embedded version of MySQL, leave the default value (BOE120).
  - If you are configuring a Microsoft SQL Server report database, leave the default value (RSDB).
  - If you are configuring an Oracle report database, leave the value empty.

- **DATABASEDB_AUDIT**
  Specifies the name of the auditing database for the Content Management Server. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:**

  - If you are installing the embedded version of MySQL, leave the default value (BOE120_AUDIT).
  - If you are configuring a Microsoft SQL Server or Oracle report database, leave the value empty.

- **DATABASEDRIVER**
  Specifies which driver to use for the report database.
  **Recommended setting:** Leave the default value. If you must change this value, use the installer to create another response file.

- **DATABASEDSN**
  Specifies the name of the ODBC connection for the report database.
  **Recommended setting:**

- If you are installing the embedded version of MySQL, leave the default value (Business Objects CMS).
- If you are using an existing instance of Microsoft SQL Server, enter the DSN.
- If you are using an exiting instance of Oracle, leave the value empty.

- **DATABASEDSN_AUDIT**
  Specifies the name of the ODBC connection for the Content Management Sever Auditing Database. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:** Leave the default value (Business Objects Audit Server).

- **DATABASENWLAYER_AUDIT**
  Specifies the Content Management Server Audit Database type. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:** Leave the default value (ODBC).

- **DATABASEPORT**
  Specifies the port on which the report database must listen.
  **Recommended setting:**
  - If you are installing the embedded version of MySQL, enter a port.
    **Default:** 3306
  - If you are configuring a Microsoft SQL Server or Oracle report database, leave the value empty.

- **DATABASEPORT_AUDIT**
  Specifies the port on which the Content Management Server Audit Database must listen. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:** A value for this parameter is not required.

- If you are installing the embedded version of MySQL, enter a port.
  **Default:** 3306
- If you are configuring a Microsoft SQL Server or Oracle report database, leave the parameter empty. CA Single Sign-On does not require that you audit the Content Management Server.

- **DATABASEPWD**
  Specifies the password of the administrator account that can access the report database.

- **DATABASEPWD_AUDIT**
  Specifies the password of the administrator account that can access the Content Management Server Auditing Database. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the

Report Server and is not used for CA Single Sign-On.
A CA Single Sign-On audit database is required to run audit–based reports.
**Recommended setting:**

- If you are installing the embedded version of MySQL, match this value with the DATABASEPWD paramter.
- If you are configuring a Microsoft SQL Server or Oracle report database, leave the parameter empty.

- **DATABASEPWD_MYSQLROOT**
  If you are installing the embedded version of MySQL, specifies the password of the MySQL root user account that can access the report database.

- **DATABASERDMS_AUDIT**
  Specifies the Content Management Server Auditing Database type. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:** Leave the default value of this parameter.

- **DATABASESERVER**
  If you are configuring an Oracle report database, specifies the name of the Oracle database service.

  **Note:** This parameter appears in a response file that configures a Microsoft SQL Server report database. Leave this value empty.

- **DATABASESERVER_AUDIT**
  Specifies the name of the Content Management Server Auditing Database server host system. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.
  A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:**

  - If you are installing the embedded version of MySQL, leave the default value (localhost).
  - If you are configuring a Microsoft SQL Server or Oracle report database, leave the parameter empty.

- **DATABASEDRIVER**
  Specifies which driver to use for the report database.
  **Recommended setting:** Leave the default value. If you must change this value, use the installer to create another response file.

- **DATABASEUID**
  Specifies the name of the administrator account that can access the report database.
  **Recommended setting:**

- If you are installing the embedded version of MySQL, enter an account name. The installer creates the administrator account with the required privileges.
- If you are configuring Microsoft SQL Server, enter an account name with database owner (DBO) privileges.
- If you are configuring Oracle, enter an account name with the following privileges enabled: create session, create table, and create procedure.
  You can also enter an account name with the CONNECT and RESOURCE roles enabled. Be sure to disable the Admin Option setting for both roles.

- **DATABASE_UID_AUDIT**
  Specifies the name of the administrator account that can access the Content Management Server Auditing Database. This parameter is not related to CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On. A CA Single Sign-On audit database is required to run audit–based reports.
  **Recommended setting:**
  - If you are installing the embedded version of MySQL, match this value with the DATABASEUID parameter.
  - If you are configuring a Microsoft SQL Server or Oracle report database, leave the value empty.

- **DATABASEUSER**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated and matches the DATABASEUID value.
  **Recommended setting:** Be sure that this value matches the DATABASEUID value.

- **DATABASEUSER_AUDIT**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated and is empty.
  **Recommended setting:** Leave the value empty.

- **DATABASE_AUDIT_CONNSVR**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** connsvr

- **ENABLELOGFILE**
  Specifies if the installer creates installation log files.
  - **0**
    Do not create log files.
  - **1**
    Create log files.

- **ENABLESERVERS**
  Specifies if servers must be enabled once the installation is complete.

- **0**
  Do not enable servers.

- **1**
  Enable servers.

**Recommended setting:** 1

- **EXPANDCMS**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated. Do not modify this parameter
  **Recommended setting:** 1

- **INSTALL.LP.EN.SELECTED**
  Specifies that the installer install the English Language pack.
  **Recommended setting:** 1

- **INSTALLDBTYPE**
  Do not modify this parameter. This parameter is automatically generated.

- **INSTALLDIR**
  Specifies the directory to which the Report Server is installed.

  
  **Note:** Be sure that you escape the backslashes in the path with a backslash character.

  **Example:** C:\\Program Files\\CA\\SC\\CommonReporting3\\

- **INSTALLLEVEL**
  Do not modify this parameter. This parameter is automatically generated.

- **INSTALLMODE**
  Specifies the installation method.

  - **New**
    Install all required server and client components.

  - **Custom**
    Select specific server and client components to install.

  - **Web Tier**
    Install only the required web application server components.

  **Recommended setting:** New

- **INSTALL_DB_TYPE**
  Do not modify this parameter. This parameter is automatically generated.

- **MYSQLPORT**
  If you are using the embedded version of MySQL, specifies the port to which MySQL must connect and listen for requests.

**Default:** 3306.
**Note:** This parameter appears in the properties file that configures Microsoft SQL Server as a report database. Leave the default value.

- **MYSQLSERVER**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated. Do not modify this parameter.
  **Recommended setting:** Leave the value empty.

- **MYSQLSERVER_AUDIT**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated. Do not modify this parameter.
  **Recommended setting:** Leave the value empty.

- **NEWCMSPASSWORD**
  Specifies the password for the default SAP BusinessObjects Enterprise administrator account. This parameter is automatically generated.
  **Recommended setting:** Do not modify this parameter. If you must change this value, use the installer to create another response file.

- **MYSQL_REMOTE_ACCESS**
  If you are using the embedded version of MySQL to function as the report database, specifies if remote access is enabled.
  **Recommended setting:** Leave the value empty.
  **Note:** This parameter appears in the properties file that configures Microsoft SQL Server as a report database. Leave the value empty.

- **NAMESERVER**
  Specifies the name of the Report Server host system.

- **NEWCMSPASSWORD**
  Specifies the password for the default SAP BusinessObjects Enterprise administrator account.
  **Recommended setting:** Do not modify this parameter. This parameter is automatically generated. If you must change this value, use the installer to create another response file.

- **NSPORT**
  Specifies the port on which the Content Management Server must listen.
  **Default:** 6400.

- **REINITIALIZE_CMS_DB**
  If you are configuring a Microsoft SQL Server or Oracle report database, specifies that the database is reinitialized.
  **Recommended setting:** 1

- **SINGLESERVER**
  Do not modify this parameter. This parameter is automatically generated. Do not modify this parameter.
  **Recommended setting:** Leave the value empty.

- **SKIP_DEPLOYMENT**
  Do not modify this parameter. This parameter is automatically generated. Do not modify this parameter.
  **Recommended setting:** Leave the value empty.

- **SelectODBCDSN**
  If you are configuring a Microsoft SQL Server or Oracle report database, this parameter is automatically generated. Do not modify this parameter.
  **Recommended setting:** Leave the value empty.

- **TOMCAT_CONNECTION_PORT**
  Specifies the port to which the embedded version of Apache Tomcat must connect and wait for requests.
  **Default:** 8080.

- **TOMCAT_REDIRECT_PORT**
  Specifies the port to which the embedded version of Apache Tomcat must redirect requests.
  **Default:** 8443

- **TOMCAT_SHUTDOWN_PORT**
  Specifies the port to which the SHUTDOWN command for the embedded version of the Apache Tomcat must be issued.
  **Default:** 8005

- **WCADOTNETINSTALL**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** False

- **WCAEXISTINGINSTALL**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** False

- **WCAJAVAINSTALL**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** False

- **WCATOMCATINSTALL**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** True

- **WDEPLOY_LANGUAGES**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** en

- **WDEPLOY_LATER**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** Leave this value blank.

- **WEBSITE_METABASE_NUMBER**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** 1

- **WEBSITE_NAME**
  If you are deploying an IIS web application server, specifies the name of the IIS website to which the SAP BusinessObjects Enterprise applications are deployed.
  CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** Default Web Site

- **WEBSITE_PORT**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** 80

## Features

This section details the parameters in the Features section of the response file.

- **REMOVE**
  Specifies which client, server, web tier, and other SAP BusinessObjects Enterprise components not to install.
  **Recommendedsetting:** Do not change the value of this parameter. To change this value, use the installer to create another response file.

- **ADDLOCAL**
  Specifies which client, server, web tier, and other SAP BusinessObjects Enterprise components to install.
  **Recommended setting:** Do not change the value of this parameter. To change this value, use the installer to create another response file.

- **ADDSOURCE**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

- **ADVERTISE**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

## BIEK

This section details the parameters in the BIEK section of the response file.

- **BIEK_INSTALL_SAMPLES**
  Specifies if the installer must install CA templates. This parameter is not related to the CA Single Sign-On reporting templates. The CA Single Sign-On Report Server Configuration wizard installs the required reporting templates. You run the wizard after installing the Report Server.

  - **0**
    Do not install sample templates.

  - **1**
    Install sample templates.

  **Recommended setting:** 0

- **SUPPRESS_REBOOT**
  Specifies if the installer reboots the computer after a successful installation.

  - **0**
    Lets the installer restart the system after a successful installation.

- **1**
  Prevents the installer from restarting the system after a successful installation. Restart the system manually to complete the installation.

**Recommended setting:** 1

# Modify the Report Server Response File for UNIX

You modify the installer response file to define installation variables. The default values reflect the information that you entered when you installed the Report Server and saved a response file.

**Important!** Consider the following items before you modify the file:

- Some parameters identify the type of database that is to function as the report database. Do not use the properties file to change the type of database that the installer is to configure. To change the report database type, use the installer to create another properties file.
- Passing a parameter in the command line overrides the respective setting in the response file.

The Report Server installer generates some parameters automatically. Do not modify the following parameters:

- PRODUCTID_NAME
- BOBJVERSION
- DATACONNECTION
- PRODUCTID_VER
- FUNCTION
- LANGUAGES_TO_INSTALL
- EXPANDSERVERS

## Manual Settings

This section details the parameter in the Manual Settings section of the response file.

- **MACHINENAME**
  Specifies the name of the Report Server host. This value overrides the local server name. If you do not provide a value, the value defaults to the local host system name.

## Paths

This section details the parameter in the Paths section of the response file.

- **BOBJEDIR**
  Specifies the path of the bobje directory. The bobje directory is automatically created in the common reporting directory.
  **Example:** /opt/CA/SharedComponents/CommonReporting3/bobje/

- **CDDIR**
  Specifies the path to the Disk1 directory in the Report Server installation kit.

- **LICENSEDIR**
  Specifies the path to the directory that contains the product license.

  **Note:** Installing the Report Server does not require you to supply a product license.

  **Recommended setting:** Leave the value empty.

## Product Information

This section details the parameters in the Product Information section of the response file.

- **BOBJELANG**
  Specifies the language setting that the installation is to use.
  **Recommended setting:** en

- **PRODUCTID_NAME**
  Specifies the name of the product that is being installed.
  **Recommended setting:** BusinessObjects

- **BOBJEVERSION**
  Specifies the version of SAP BusinessObjects Enterprise.
  **Recommended setting:** 12.0

- **PRODUCTID_VER**
  Specifies the version of the product being installed.
  **Recommended setting:** 12.3

- **BOBJELICENSEKEY**
  Specifies the license key that is required to install the product. This value appears encrypted.
  **Recommended setting:** Do not modify this value.

- **PIDKEY**
  Specifies the product ID key. This value appears encrypted.
  **Recommended setting:** Do not modify this value.

## Installation Information

This section details the parameters in the Installation Information section of the response file.

- **FUNCTION**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** install

- **INSTALLTYPE**
  Specifies the installation method.

  - **new**
    Install all required server and client components.

  - **custom**
    Select specific server and client components to install.

  - **webtier**
    Install only the required web application server components.

  **Recommended setting:** new

- **INSTALLMODE**
  Specifies a comma–delimited list for the Report Server installation operating modes. This parameter supports the following options:

  - install

  - modify

  - remove

  - interactive

  **Recommended setting:** install

- **LOCALNAMESERVER**
  Specifies the name of the Report Server host system.

- **BOBJEINSTALLLOCAL**
  Specifies whether to execute a user or system installation.
  **Recommended setting:** user

- **LANGPACKS_TO_INSTALL**
  Specifies the language packs to install. CA Single Sign-On only supports the English language pack. The installer installs the English language pack by default.
  **Recommended setting:** Leave the value empty.

- **LANGUAGES_TO_INSTALL**
  Specifies all languages included in SAP BusinessObjects Enterprise. These values represent the available languages, not the language packs to install.
  **Recommended setting:** Leave the comma–delimited list of values.

- **BOBJEUSERNAME**
  Specifies the name of the non–root user account.
  **Recommended setting:** Be sure that this value matches the value of BIEK_INSTALL_USER in the BIEK section of the response file.

- **EXPANDSERVERS**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** Leave the value empty.

## Tomcat

This section details the parameters in the Tomcat section of the response file.

- **INSTALLTOMCAT**
  Specifies if the embedded version of Apache Tomcat must be installed.
  **Recommended setting:** yes
  **Note:** Although the installer lets you configure an existing instance of a web application server, CA Single Sign-On only supports the embedded version of Apache Tomcat.

- **CONNECTORPORT**
  Specifies the port to which the embedded version of Apache Tomcat must connect and wait for requests.
  **Default:** 8080.

- **REDIRECTPORT**
  Specifies the port to which the embedded version of Apache Tomcat must redirect requests.
  **Default:** 8443

- **SHUTDOWNPORT**
  Specifies the port to which the SHUTDOWN command for the embedded version of the Apache Tomcat must be issued.
  **Default:** 8005

## Application Server

This section details the parameters in the Application Server section of the response file.

- **AS_DIR**
  Specifies the path to which the embedded version of Apache Tomcat is installed. The path is automatically set using the installation directory.
  **Example:** /opt/CA/SharedComponents/CommonReporting3//bobje/tomcat/

- **AS_SERVER**
  Specifies the type of Java web application server to deploy. If the embedded version of Apache Tomcat is installed, the value defaults to tomcat55. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** tomcat55

- **AS_INSTANCE**
  Specifies the name of the web application server instance. If the embedded version of Apache Tomcat is installed, the value defaults to localhost. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** localhost

- **AS_VIRTUAL_HOST**
  If you are deploying the Report Server to a virtualized environment, specifies the virtual host to which the application must be bound.
  **Recommended setting:** Leave the value empty.

- **AS_ADMIN_PORT**
  Specifies the port on which the web application server is listening. If the embedded version of Apache Tomcat is installed, the value is empty. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** Leave the value empty.

- **AS_ADMIN_USERNAME**
  Specifies the account name of the administrator account that is used to access the web application server. If the embedded version of Apache Tomcat is installed, the value is empty. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** Leave the value empty.

- **AS_ADMIN_PASSWORD**
  Specifies the password of the administrator account that can access the web application server. If the embedded version of Apache Tomcat is installed, the value is empty. CA Single Sign-On only supports the embedded version of Apache Tomcat.
  **Recommended setting:** Leave the value empty.

- **AS_ADMIN_SECURE**
  Specifies that an administrator credential must be passed to access the web application server. This parameter applies to a web application server that CA Single Sign-On does not support.
  **Recommended setting:** false

- **AS_APPSERVER_ID**
  Specifies the name of the application server. This parameter applies to a web application server that CA Single Sign-On does not support.
  **Recommended setting:** Leave the value empty.

- **AS_GROUP_ID**
  Specifies the group ID of the application server. This parameter applies to a web application server that CA Single Sign-On does not support.
  **Recommended setting:** Leave the value empty.

- **WEBDEPLOYACTION**
  Specifies the action to perform on the web application server.
  **Recommended setting:** deploy

- **REDEPLOYWEBAPPS**
  Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** true

## CMS Cluster

This section details the parameters in the CMS Cluster section of the response file.

- **CMSCLUSTER**
  Specifies if you are adding servers to an existing Content Management Server.
  **Recommended setting:** no

- **CLUSTER_NAMESERVER**
  If you are clustering servers to a Content Management Server, specifies the name of the Content Management Server.
  **Recommended setting:** Leave the value empty.

- **CLUSTERPORTNUMBER**
  If you are clustering servers to a Content Management Server, specifies the port on which the Content Management Server is listening. This value defaults to the value of the CMSPORTNUMBER parameter. The parameter is located in the CMS section of the response file.
  **Recommended setting:** Be sure that this value matches the value of CMSPORTNUMBER.

## CMS

This section details the parameters in the CMS section of the response file.

- **DBTYPE**
  Specifies the type of database that is to function as the report database.
  **Recommended settings**

  - MySQL

  **Important!** CA Single Sign-On only supports the embedded version of MySQL.

  - Oracle

  **Important!** The Report Server is a common component that CA products share. As such, the installer lets you configure database types and versions that CA Single Sign-On does not support but other CA products do. For a list of supported databases, see the Platform Support Matrix.

- **SERVICENAME**
  Specifies the service name of the Oracle database functioning as the report database.
  **Note:** If you are configuring MySQL, this value defaults to BOE120. Leave the default value.

- **DATABASEUID**
  Specifies the name of the administrator account that is used to access the report database.
  **Recommended setting:**

- If you are installing the embedded version of MySQL, enter an account name. The installer creates the administrator account with the required privileges.
  - If you are configuring Oracle, enter an account name with the following privileges enabled: create session, create table, and create procedure.
    You can also enter an account name with the CONNECT and RESOURCE roles enabled. Be sure to disable the Admin Option setting for both roles.

- **DATABASEPASSWORD**
  Specifies the password of the administrator account that is used to access the report database.

- **CMSNAMESERVER**
  Specifies the name of the Report Server host system.
  **Recommended setting:** Be sure that this value matches the value of LOCALNAMESERVER in the Installation Information section of the response file.

- **CMSPORTNUMBER**
  Specifies the port to which the Content Management Server must connect and wait for requests.
  **Default:** 6400

- **CMSPASSWORD**
  Specifies the password for the administrator account that is to access the Central Management Server. This password is for the SAP BusinessObjects Enterprise system administrator account.

- **SIANODENAME**
  Specifies the name of the Service Intelligence Agent (SIA) node.

  **Note:** Do not use spaces or non–alphanumeric characters.

- **SIAPORTNUMBER**
  Specifies the port to which the SIA must connect and wait for requests.
  **Default:** 6410

- **REINIT**
  Specifies if the report database must be reinitialized. Do not modify this parameter. This parameter is automatically generated.
  **Recommended setting:** yes

## MySQL

This section details the parameters in the MySQL section of the response file.

- **INSTALLMYSQL**
  Specifies if the embedded version of MySQL must be installed.
  **Recommended setting:**
  - If you are installing the embedded version of MySQL, enter yes.
  - If you are configuring an Oracle report database, leave the value empty.

- **SERVICEPORT**
  If you are installing the embedded version of MySQL, specifies the port to which MySQL must connect and listen for requests.
  **Default:** 3306

  > **Note:** If you are configuring an Oracle report database, leave the default value.

- **MYSQLHOSTNAME**
  If you are installing the embedded version of MySQL, specifies the IP address of the MySQL host system. The MySQL host system is the same as the Report Server host system.

- **MYSQLROOTPASSWORD**
  If you are installing the embedded version of MySQL, specifies the password of the MySQL root user account.

## Audit

This section details the parameters in the Audit section of the response file.

These parameters are not related to the CA Single Sign-On audit–based reports. The Content Management Server Auditing Database is used to audit activities specific to the Report Server and is not used for CA Single Sign-On.

A CA Single Sign-On audit database is required to run audit–based reports.

- **AUDITINGENABLED**
  Specifies if the Content Management Server Auditing Database is enabled.

  - If you are installing the embedded version of MySQL, the Auditing Database is automatically enabled.
    **Recommended setting:** yes

  - If you are configuring an Oracle report database, auditing the Content Management Server is not required.
    **Recommended setting:** no

- **SERVICENAME_AUDIT**
  Specifies the name of the audit service that the Content Management Server uses. This value defaults to BOE120_AUDIT, even if you are not enabling the Audit Database.
  **Recommended setting:** Leave the default value.

- **SERVICEPORT_AUDIT**
  Specifies the port number to which the Content Management Server Audit Database must connect and listen for requests. This value defaults to 3306.
  **Recommended setting:**

  - If you are installing the embedded version of MySQL, match this value to the SERVICEPORT parameter in the MySQL section of the response file.

  - If you are configuring an Oracle report database, leave the default value.

- **MYSQLHOSTNAME_AUDIT**
  Specifies the IP address of Content Management Server Audit Database host system.
  **Recommended setting:**
  - If you are installing the embedded version of MySQL, match this value to the MYSQLHOSTNAME parameter in the MySQL section of the response file.
  - If you are configuring an Oracle report database, leave the value empty.
- **DATABASEUID_AUDIT**
  Specifies the name of the administrator account that is used to access the Content Management Server Auditing Database.
  **Recommended settings:**
  - If you are installing the embedded version of MySQL, match this value to the DATABASEUID parameter in the MySQL section of the response file.
  - If you are configuring an Oracle report database, leave the value empty.
- **DATABASEPWD_AUDIT**
  Specifies the password of the administrator account that is used to access the Content Management Server Auditing Database.
  **Recommended settings:**
  - If you are installing the embedded version of MySQL, match this value to the MYSQLROOTPWD parameter in the MySQL section of the response file.
  - If you are configuring an Oracle report database, leave the value empty.

## Marketing Products

This section details the parameters in the Marketing Products section of the response file.

- **ENABLEMP**
  Specifies which client, server, web tier, and other SAP BusinessObjects Enterprise to enable manually.
  **Recommended setting:** Leave the value empty.
- **DISABLEMP**
  Specifies which client, server, web tier, and other SAP BusinessObjects Enterprise components disable manually.
  **Recommended setting:** Leave the value of this parameter empty.

## BIEK

This section details the parameters in the BIEK section of the response file.

- **BIEK_INSTALL_USER**
  Specifies the non–root user account that the installer must use.
- **BIEK_INSTALL_GROUP**
  Specifies the group to which the non–root user belongs.

- **BIEK_CASHCOMP**
  Specifies the full path to which the CA Shared Components environment variable (CASHComp) must be set. If CASHComp is already set, the installer ignores this value.

- **BIEK_INSTALL_SAMPLES**
  Specifies if the installer must install CA templates. This parameter is not related to the CA Single Sign-On reporting templates. The CA Single Sign-On Report Server Configuration wizard installs the required reporting templates. You run the wizard after installing the Report Server.

  - **0**
    Do not install sample templates.

  - **1**
    Install sample templates.

  **Recommended setting:** 0

- **BIEK_MIGRATE_CMS_DATA**
  Do not modify this parameter. This parameter is for upgrades only.
  **Recommended setting:** 0

- **BIEK_SOURCE_CMS_PASSWORD**
  Do not modify this parameter. This parameter is for upgrades only.
  **Recommended setting**: Leave the value empty.

# Modify the CA Single Sign-On Report Server Configuration Wizard Properties File

You modify the Report Server Configuration Wizard properties file to define configuration variables. The default parameters, passwords, and paths in this file reflect the information entered the last time the wizard ran.

The properties file has the following parameters:

- **DEFAULT_BOXI_INSTALL_PATH**
  Specifies the Report Server installation path.

- **DEFAULT_BOXI_PASSWORD**
  Specifies the password of the default BusinessObjects administrator account.

  **Note:** If you are using encrypted value in ENCRYPTED_BOXI_PASSWORD, this parameter does not require a value.

- **ENCRYPTED_BOXI_PASSWORD**
  Specifies the encrypted password of the default BusinessObjects administrator account.

**Important!** Do not change the encrypted value. To enter another password, comment the encrypted password and specify a value for DEFAULT_BOXI_PASSWORD.

- **DEFAULT_AUDIT_DATABASE_TYPE**
  Specifies the type of database functioning as the CA Single Sign-On audit store.

  **Note:** You do not have to configure a CA Single Sign-On audit database before running the Report Server Configuration Wizard.

  **Values:** 1 or 2

  1 – Specifies Microsoft SQL Server.

  2 – Specifies Oracle.

# Run the Report Server Installer

You run an unattended installation to install the Report Server without user interaction.

## Before You Install

Consider the following items before installing the Report Server:

- You install the Report Server using the installation media on the Technical Support site.

For a list of installation media names, see the *Policy Server Release Notes*.

- The Report Server is compiled as 32–bit native binary and is designed to use 32–bit data source middleware connectivity. If you are installing to a Windows 64–bit operating system, be sure to create the DSN using odbcad32.exe. This executable is located in the following location: *install_home*\Windows\SysWOW64.

  - **install_home**
    Specifies the installation path of the Windows operating system.

- The Report Server installation includes the following components that run as processes:

  - The Content Management Server

  - The Server Intelligence Agent

  These components require TCP/IP ports to communicate. The installer lets you modify the default settings to prevent port conflicts on the Report Server host system.

- **Important!** The installation zip contains multiple folders. The Report Server installer requires this folder structure. If you moved the Reports Server installer after extracting the zip, copy the entire folder structure to the same location. Be sure that you execute the installation media from the folder structure.

## Windows

**Follow these steps:**

1. Exit all applications that are running.
2. Open a command prompt.
3. Change the directory to *temporary_location*.
   - ***temporary_location***
     Specifies the location to which you copied the installation media.
4. Enter the following command:

   ```
   installation_media silent path_to_response_file
   ```

   **Important!** If User Account Control (UAC) is enabled in Windows Server, open the command-line window with administrator permissions. Open the command-line window this way even if your account has administrator privileges.

   - ***installation_media***
     Specifies the name of the Report Server installation executable.

For a list of installation media names, see the *Policy Server Release Notes*.

***path_to_response_file***

Specifies the path to the file. The path must include the response file name.

**Note:** The response file does not have to be in the same directory as the installation executable.

The silent installation begins.

## UNIX

**Follow these steps:**

1. Exit all applications that are running.

2. Be sure that you are using an account with root–user privileges.

3. Open a Bourne shell and navigate to *temporary_location*.

   - **temporary_location**
     Specifies the location to which you copied the installation media.

4. Enter the following command:

   ```
   ./installation_media silent path_to_response_file
   ```

   - ***installation_media***
     Specifies the name of the Report Server executable.

For a list of installation media names, see the *Policy Server Release Notes*.

- ***path_to_response_file***
  Specifies the path to the response file. The path must include the response file name.

**Note:** The response file does not have to be in the same directory as the installation executable.

The silent installation begins.

# Install the Report Templates

You run an unattended installation to install the CA Single Sign-On report templates without user interaction.

## Before You Install

You install the CA Single Sign-On report templates using the installation media on the Technical Support site.For a list of installation media names, see the *Policy Server Release Notes*.

# Install and Configure CA SiteMinder® SPS Silently

After you have installed and configured CA Access Gateway for the first-time, you can reinstall it unattended at a later time or install another instance of CA Access Gateway unattended using saved configuration data.

After installation, CA Access Gateway creates a sample properties file in the sps-home /install_config_info folder. After configuration, the same properties file is updated with additional properties for configuration. This properties file is used for subsequent silent installation and configuration with customized values.

**Follow these steps:**

1. Open a command window.
2. Navigate to the folder where you installed the properties file. The default is sps-home /install_config_info.
3. Open the command prompt.
4. Perform one or both the following steps:
    a. Execute the following command to perform a silent installation:

        ```
        ca-proxy-version-operating_system -i silent -f ca-sps-installer.
        properties
        ```

    b. Execute the following command to perform a silent configuration:

        ```
        ca-sps-config -i silent -f ca-sps-installer.properties
        ```

    The installation or configuration proceeds without further user interaction.