CA SSO

Agent for Siebel Guide

r12.51

CA Technologies Product References

This document references the following CA Technologies products:

- CA SSO

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Overview and Architecture

This section contains the following topics:

Background

The Web is becoming the standard interface for newly deployed applications. In an effort to meet the requirements of customers and to enable more widespread use of applications, many leading ERP vendors, including Siebel, have developed either web-based versions of applications or web-based front ends for applications. As a minimum, these web-based front ends provide:

- A standard look and feel for employees
- User authentication
- Basic security (user identity and password)
- Single sign-on (SSO) capability

CA SSO lets you create a centrally managed environment, providing a secure, personalized user experience across all web applications. Through published interfaces, CA SSO can authenticate users to Siebel. This integration enables the Siebel .COM-based applications to coexist with other portals and web applications, while offering the maximum user experience and benefit.

Increased Security with Tier 2 Integration

Tier 1 Integration

Tier 1 integration typically describes the process in which an underlying application reads and interprets the authentication information passed by CA SSO so the application (Siebel) can log the user in and create its own session if necessary. Tier 1 integration is the minimum security required to provide SSO. With Tier 1 integration, the underlying application fully trusts that the information was sent from CA SSO and does no verification. Tier 1 integration can leave important integration issues untouched, such as session timeouts. In Tier 1, the point of trust is entirely within the first tier—the web server. This design is adequate in environments where the application server and web server are located entirely on a trusted network, where security requirements are low to moderate.

Tier 2 Integration

In Tier 2 implementation, the point of trust moves away from the web server and into a more trusted host, in this case the Siebel Object Manager.

In Tier 2 integrations, the application that implements the application logic and security is given the ability to call CA SSO APIs to communicate with a Policy server, to validate the information that is presented, ostensibly, from the web agent. The API used in this integration is a Siebel-specific API called the Security Adapter API.

Session Linking

Many web-based applications use an independent session management scheme, frequently through the use of a cookie. Therefore, replay prevention and session management logic of CA SSO may be bypassed. The possibility that the CA SSO and application sessions could lose synchronization with each other is one of the main security problems when integrating applications that maintain their own sessions.

Due to the enormous value of the data stored in Siebel, CA believes that extra security measures are warranted. The integration documented here includes the SessionLinker component, whose purpose is to prevent such session synchronization issues. SessionLinker is a web server plug-in that monitors the CA SSO Session ID header and Siebel session cookie. When the two sessions diverge, action is taken to prevent the application from operating until a new session within Siebel is established. By default the action is to destroy the Siebel session, which causes Siebel to create a new session for the correct user.

Note: For more information about SessionLinker, see the *SessionLinker Guide*.on the CA Support site.

Architecture

Components

Following are the main components of this product:

- An Active Response that generates an “Authentication Ticket” securely identifying the user.
- Siebel templates and a session initiator page.
- A CA SSO-enabled Siebel Security Adapter conforming to Siebel’s Security Adapter API.

- The web server code used for initiating a Siebel session through the Security Adapter.
- An authentication scheme that accepts the Authentication tickets generated by the Active Response.
- SessionLinker, which is a web server plug-in that maintains a mapping from the CA SSO session to the Siebel session to prevent session hijacking attacks.

  Note: For more information, see the *SessionLinker Guide*.

Conventional Environment

In a conventional Siebel COM environment, users connect to a web server, which in turn connects to the Siebel Server. The web server collects the user's credentials and transports them to the Siebel Server, which validates them, generates a session cookie, and outputs HTML content to the user. See the following illustration:

Integrated Environment

The flow changes after CA SSO and this product are added to the environment. Before reaching the Siebel Web Engine on the web server, the web agent either collects and verifies the user's credentials or verifies an existing CA SSO session by communicating with the Policy server. A single sign-on ticket is generated and passed through the Siebel server in place of the user's password. The web agent allows the request to be passed on to the Siebel Web Engine.

The Siebel Web Engine passes the request along to the Siebel server – including the username and ticket (in place of a password). The Siebel server, via a customized Security Adapter, communicates with the Policy Server to verify the username and ticket.

In an integrated environment, user authorization and data access within Siebel continue to operate exactly as they had without the SSO agent.

Thick Clients

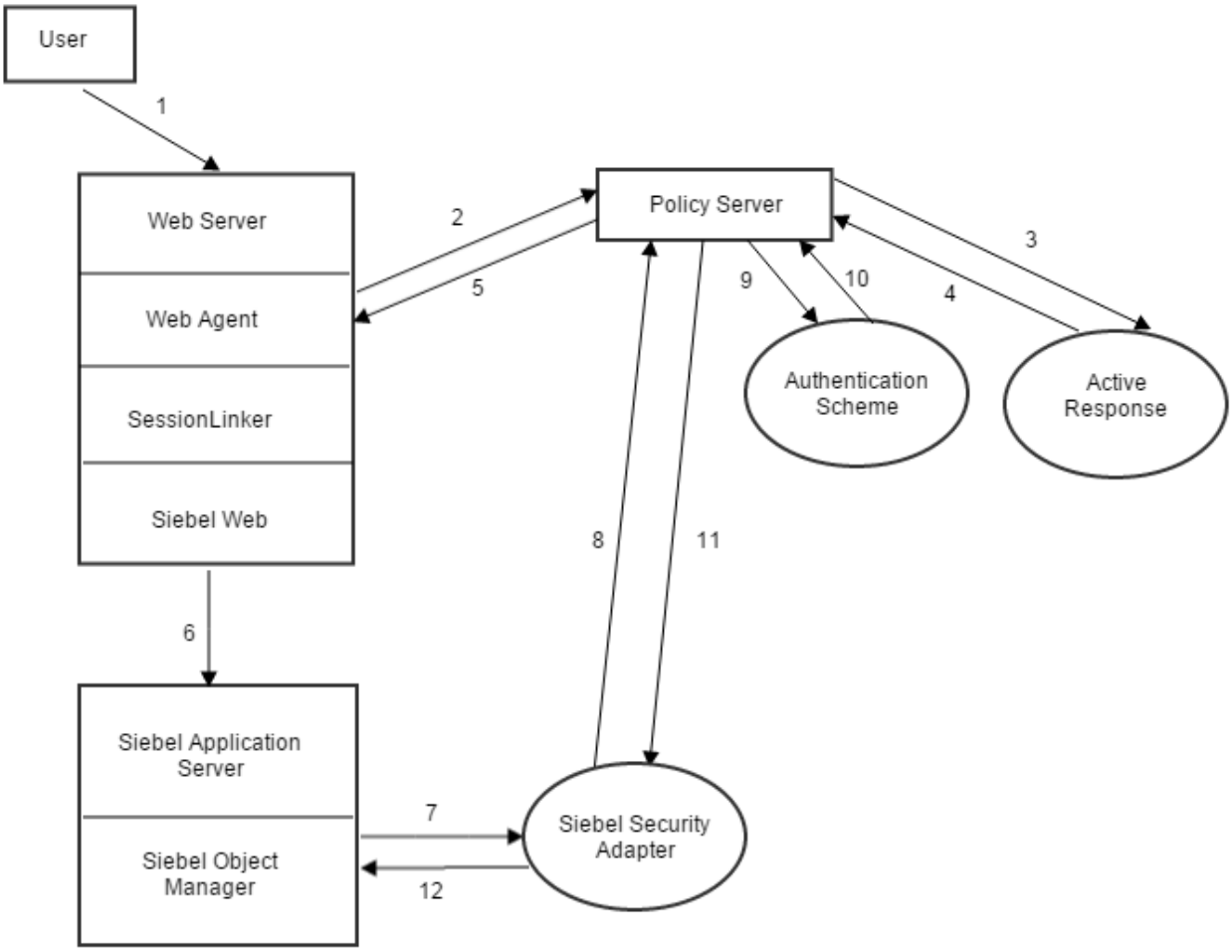
In addition to single sign-on through the web, this product supports conventional Siebel thick clients authenticating with username and passwords through the Siebel server. Once this product is installed, user passwords stored in the database are no longer accepted for authentication; instead users need to present their CA SSO username and password.

Note: Through a number of means, Siebel is enabled to accept the CA SSO username and password as well as the database username and password. To enable this support, you will need to configure the Policy Server to authenticate users out of both the enterprise directory and the Siebel database. Contact Technical Support for further information.

Data Flow

The following illustration shows the various components within an integrated CA SSO/Siebel installation in more detail.

Note: Certain components are omitted – for example the user directory, the Siebel database, and additional CA SSO responses are not shown, nor are the normal IsProtected, Login, and IsAuthorized calls shown.



The steps in the preceding illustration are the following:

1. User makes a request to the Siebel application, for example, http://*machine.domain:port*/service_enu.
2. Web Agent intercepts the request, and uses Policy Server to perform Authentication/Authorization.

   Note: If the Siebel responses are configured for a Get-Post rule under the realm protecting the Siebel application in the Policy Server, steps 3, 4 and 5 are implemented. Otherwise, control passes to step 6.

3. If the authentication by Policy Server is successful, the following takes place:
   - Active Response is fired, and generates the Siebel Authentication Ticket, SIEBELTICKET. This authentication ticket is specific to the user accessing the application.
   - Siebel user response is fired, sending a user attribute, whose value maps to a valid Siebel user.
4. Relevant results are provided to the Policy Server.
5. Web Agent receives the Active response and Siebel user response from Policy Server, and generates the HTTP headers for these responses: HTTP_SIEBELTICKET and HTTP_SIEBELUSER. These responses are sent to the session launching code.
6. The Siebel web component (SWSE) intercepts the request, performs some Siebel-controlled request transformation, and sends the request to Siebel Object Manager using Anonymous user credentials.
   a. On receiving the request from SWSE, Siebel Object Manager checks for the Anonymous user credentials before sending the Siebel Login Web template (SWELogin.swt) customized with the Siebel Agent session launching code back to the SWSE. For verifying the anonymous user credentials, Object manager calls Security Adapter (or Security Provider).

      Security Adapter verifies the passed Anonymous user credentials against the Anonymous username/password credentials specified in the SmSiebelSSO.conf file.

      Once the Anonymous user credentials are successfully verified, the customized SWELogin.swt is fetched and sent to SWSE.

   b. SWSE converts the modified SWELogin.swt into HTML format, and redirects the request to the /SiebelConnector/siebelstartup.asp hosted at the web server. A typical URL format for such a redirection is as follows:

      http://<machine.domainname:Port/SiebelConnector/siebelstartup.asp?URL=http%3A//corsairerp.ca.com/service_enu/start.swe%3FSWECmd%3DStart%26SWEHo%3Dcorsairerp.ca.com

   c. Since the /SiebelConnector/* resources are protected by Policy Server using Siebel SSO Authentication scheme, the web agent validates the user session from the Policy Server.

      Step 3 is carried out, and the Siebel Authentication Ticket (SIEBELTICKET) and SIEBELUSER responses are generated. Web Agent receives the above responses and generates HTTP headers HTTP_SIEBELUSER and HTTP_SIEBELTICKET from them.

      Important! The above two responses get fired irrespective of the fact that they have or have not got fired in the previous step 3. The values generated by these responses in the above two steps are used in the subsequent steps.

d. siebelstartup.asp converts the URL into a form which SWSE uses to send the username and password to Siebel Object Manager.

   siebelstartup.asp extracts username (SIEBELUSER) and password (Siebel Ticket) from the HTTP request headers, HTTP_SIEBELUSER and HTTP_SIEBELTICKET, and places their values in 'SWEUsername'and 'SWEPassword' parameters respectively.

7. SWSE again sends the request to Siebel Object manager along with the above user credentials. Siebel Object Manager calls Security Adapter or Security Provider and passes the user credentials to it for verification.
8. Security Provider contacts Policy Server and accesses the protected resource /SiebelConnector/ (configured in the SmSiebelSSO.conf file) using the user credentials previously received (SIEBELUSER and SIEBELTICKET).
9. Policy Server uses the Siebel SSO authentication scheme to verify the user credentials.

   Authentication Scheme verifies the user credentials, SIEBELUSER, and the Siebel authentication Ticket (SIEBELTICKET).
10. The Siebel SSO Authentication scheme results are returned to Policy Server.
11. Policy Server returns the results of user credentials authentication back to Security Adapter. These results also contain relevant information, such as Siebel Roles and Siebel User Response.
12. Security provider checks the SIEBELUSER response returned previously against the response that was extracted from the HTTP headers. If the user credential authentication is successful and Siebel user responses match, Security Provider reports the results to Siebel Object Manager and creates a Siebel user context for the SIEBELUSER user, which creates a Siebel user session and sends the request to the main application startup page in the SWSE.

More information:

Monitoring the Processing of a Request (see page 57)

Chapter 2: Single Sign-On Security Zones

This section contains the following topics:

SSO Security Zones and CA SSO Agent for Siebel

This product supports SSO Security Zones. The SSO Security Zones are defined and managed by the Web Agent.

Note: For more information about configuring SSO Security Zones, see the CA SSO documentation.

Users have the ability to define single sign-on security zones within the same cookie domain, representing a single zone, or across multiple cookie domains, representing different zones. As a result, users have single sign-on within the same zone, but may be re-challenged when entering a different zone, depending on the trust relationship defined between the zones. Zones included in a trusted relationship will not rechallenge a user that has a valid session in any zone in the group.

Single sign-on security zones are implemented entirely by Web Agents. Each zone must reside on a separate Web Agent instance. Multiple zones cannot be created on the same Agent instance.

A security zone is identified by cookies generated by the Web Agent. By default, the Web Agent generates two cookies, a session cookie named SMSESSION and an identity cookie named SMIDENTITY. When you configure security zones, the Web Agent can generate session cookies and identity cookies with unique names so that the zone affiliation is reflected in the cookie names.

Security Zones Benefits

The SSO Security Zones feature is intended for use in situations where CA SSO administrators wish to segment their single sign-on environments within the same cookie domain. For example, consider the CA.COM domain. Under standard CA Single Sign-On functionality, all CA SSO protected applications in CA.COM would use the cookie SMSESSION to manage single sign-on. Consider the following scenario in which SSO Security Zones do not exist:

1. A user accesses an application (APP1). The user is challenged by CA SSO, logs in to CA SSO, and creates an SMSESSION cookie.
2. The user accesses a second application (APP2) and is once again challenged by CA SSO. (Rules prevent SSO from occurring because the user does not have access to APP2 using the APP1 user credentials.) The user logs in and creates a new SMSESSION cookie overwriting the old one with the new logged in session for APP2.
3. The user now returns to APP1 and is challenged yet again, since the user lost the original APP1 session and the APP2 session might not be accepted for APP1. Therefore, SSO does not occur between APP1 and APP2, causing a very frustrating situation.

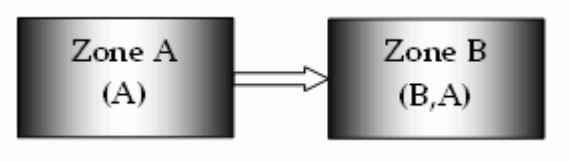
With SSO Security Zones, APP1 can be placed in zone Z1 and APP2 can be placed in zone Z2. Now logging into APP1 creates a Z1SESSION cookie and access to APP2 results in a Z2SESSION cookie. With different names, the cookies no longer overwrite each other so there is only one login per application now, not one for each time the user moves between the applications as in the example above.

Prior to the SSO Security Zones feature, the only way to perform the same grouping of SSO for applications was to create different network domains and therefore different cookie domains (CA1.COM, CA2.COM, and so on), and use various multi-cookie domain configurations with cookie providers. This is not desirable in most enterprises, since using multiple network domains has certain IT maintenance and support consequences.

Security Zone Basic Use Case

Single sign-on can, on a controlled basis, be broken into several security zones that have configurable trust relationships. For example, consider Zone A and Zone B:

- Zone A has only one trusted zone, its own Zone A.
- Zone B has two trusted zones, its own Zone B as well as Zone A.



The trust relationship in the above illustration is indicated by the arrow, meaning that the user sessions established in Zone A can be used for single sign-on in Zone B.

In this example, Zone A might be an administrator-only zone, while Zone B might be a common access zone. An administrator authenticated in Zone A gains access to Zone B without being rechallenged. However, a user authenticated in Zone B is re-challenged when trying to access Zone A.

User sessions in different zones are independent of each other. Suppose a user authenticates in Zone A first, and then authenticates again in Zone B. Two different sessions are created. In fact, the user may have different identities in both sessions. When the user returns to Zone A, the session established in that zone is used.

Consider what would happen if a user is validated using single sign-on in a zone where that user does not yet have a session. If the user authenticates in Zone A and then visits Zone B for the first time, then a user session is created in Zone B, based on the session information in Zone A, possibly updated by the Policy Server. Note that the user session in Zone A is not updated until the user returns to Zone A.

Chapter 3: Pre-Installation Steps

This section contains the following topics:

System Requirements

Following are the minimum requirements for using this product:

- Policy Server
- A web server with a web agent
- Siebel server
- Siebel COM components
- SessionLinker web server plug-in

For updated information about platform and web server support, see the appropriate Platform Support Matrix available in the CA Support website.

SessionLinker

To ensure security, SessionLinker must be installed. Although this product provides single sign-on to Siebel without SessionLinker, unless SessionLinker is installed, the integration is not secure.

SessionLinker prevents session synchronization issues by monitoring the CA SSO Session ID header and the Siebel session cookie sent by the user. When the two sessions diverge, action is taken to prevent the application from operating until a new session within Siebel is established. By default, the action is to destroy the existing session, which forces Siebel to create a new session for the correct user. Another possibility is not to destroy the existing session, but instead, to redirect the user to a configured redirect URL.

Note: The configuration parameter for SessionLinker is COOKIE=_sn.

Selecting and Configuring Database Credentials

Once an external authentication system such as CA SSO is implemented, Siebel is no longer capable of employing the individual user's credentials to connect to the database for the following reasons:

- CA SSO does not store or expose the user's credentials once the user has been authenticated. This is intentional for security reasons.
- Even if CA SSO stored the user's credentials, there is no way to know or guarantee that the database would be able to use those credentials – users might authenticate to CA SSO with certificates, SecurID or other one-time passwords, NTLM or some other authentication scheme which would not be acceptable to the database.

The Siebel Object Manager continues to communicate with the database for all data; however, because users no longer present credentials that the Object Manager can use to connect on their behalf, a special administrative account is necessary. This account's credentials need not be published, and are not used by any person or application other than the Siebel Object Manager.

The use of a generic database user does not in any way impair the ability to audit user activity because Siebel's internal access control, data protection, and audit capabilities continue to operate as with individual user database accounts. A database account should be created and the password set to a complex, non-guessable value.

A benefit of Siebel using a generic database account is that after this product is installed, individual database accounts are no longer necessary. This relieves the system of the administrative burden of account creation, password maintenance or synchronization, and removal upon termination of employment.

Chapter 4: Installing and Configuring CA SSO Agent for Siebel

This section contains the following topics:

Gather Information for the Installation Wizard

The installation wizard requires the following information:

Select the Components

Specifies the components that were previiously installed on the host where you are installing the product.

Values: Policy Server, Web Agent, Siebel Server

Install Folder

Specifies the directory where the product files are installed.

Default:

Windows:

```
C:\Program Files\CA\siebel
```

UNIX:

```
<home-dir>/CA/siebel
```

Run the Installation Wizard on Windows

Important! Before you run the installation wizard, verify that you have installed the required components.

If you are running the installer on the host that has the Siebel server installed, then configure the product to complete the installation.

Note: We recommend that you quit all other programs before you start the installation.

Follow these steps:

1. Double-click the following installation executable file:

   ca-erp-siebel-<*version*>-<*operating_environmentprocessor_type*>.exe

   Note: To install using console, open a console window and then run the previous command with the -i console option.

   Note: To install unattended, open a console window and then run the previous command with the "-i silent –f <installer_properties_file>" options. The installer properties file (ca-siebel-installer.properties) is located in the install_config_info directory of the product.

   Important! To install unattended, you must install the product using wizard or console once. The installer properties file is required for unattended installation. Before you perform the unattended installation, verify that the installer properties file is updated based on the host where you install the product.

2. Click Next.
3. Accept the License Agreement and click Next.
4. Select the components that are already installed on the host where you are installing the product and click Next.
   - Siebel application server
   - Web server, where the Web Agent has been installed.
   - Policy Server

   Note: Depending on your configuration, whether the selected servers are on the same or on different machines, you must run the Installer once, twice or three times.
5. Specify the location where you want to install the product.

   Default: C:\Program Files (x86)\CA\siebel
6. Review your selections, and click Install.

   If you are running the installer on the host that has the Siebel server installed, the installer prompts you to configure the product.
7. Complete the configuration wizard.

More information:

Gather Information for the Configuration Wizard (see page 25)

Run the Installation Wizard on UNIX

Important! Before you start the installation, make sure that you have installed the required components.

If you are running the installer on the host that has the Siebel server installed, then configure the agent to complete the installation.

Note: We recommend that you quit all other programs before you start the installation.

Follow these steps:

1. Execute the following file:

   `ca-erp-siebel-<version>-<operating_environmentprocessor_type>.bin`

   Note: To install using console, open a console window and then run the previous command with the -i console option.

   Note: To install unattended, open a console window and then run the previous command with the "-i silent –f <installer_properties_file>" options. The installer properties file (ca-siebel-installer.properties) is located in the install_config_info directory of the product.

   Important! To install unattended, you must install the product using wizard or console once. The installer properties file is required for unattended installation. Before you perform the unattended installation, verify that the installer properties file is updated based on the host where you install the product.

2. Follow the instructions and complete the installation.

Gather Information for the Configuration Wizard

Before you run the configuration wizard, gather the following information required for the installer:

Admin User Name and Password

Specifies the name and password of an administrator who has the right to register a trusted host with the Policy Server. The name that you provide here must match the name of an administrator defined at the Policy Server

Enable Shared Secret Rollover

Specifies that the Policy Server generates a new shared secret periodically, which is used to encrypt the communication to this product.

Trusted Host Name

Specifies the name of the host that you want to register with the Policy Server.

Host Configuration Object

Specifies the name of the host configuration object that is already defined at the Policy Server.

Note: For configuring Policy Server Clusters, see the CA SSO documentation.

Policy Server IP Address

Specifies the Policy server IP address with which the product communicates when validating sessions.

FIPS Encryption Mode

Determines whether the agent communicates with the Policy Server using the 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 agents read and write information using only FIPS 140-2 algorithms.

Important! A CA SSO 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.

Resource

Specifies a resource protected with the Siebel authentication scheme.

Example:

```
/SiebelConnector/
```

Action

Specifies the desired action on the Resource.

Values: GET, POST

Default: GET

Agent Name

Specifies the name of the agent with which the resource specified earlier is protected.

Log File

(Optional) Specifies the full path to the log file. Make sure the folder containing the log file exists.

Example

Windows:

```
C:\logs\connector.log
```

Log Level

(Optional) Specifies the level of logging.

Values: 0 - No logging, 1 - Errors, 2 - Information, 3 - Debug

Default: 0

For production, we recommend that you set LogLevel to 1.

Database User Name and Password

Specifies the credentials of a generic user that is used to connect to the database.

Example: SADMIN

Anonymous User Name and Password

Specifies the credentials of an anonymous Siebel user.

Siebel Response User Attribute Name

Specifies the response user attribute configured in the Policy Server whose value maps to a valid Siebel user.

Default: SIEBELUSER

Run the Configuration Wizard

Use the configuration wizard to configure the product.

Follow these steps:

1. Run the appropriate file for your operating environment:

   Windows:

   ```
   <agent_install_dir>\install_config_info\ca-siebel-config.exe
   ```

   UNIX:

   ```
   <agent_install_dir>/install_config_info/ca-siebel-config.bin
   ```

   Note: To configure using console, open a console window and then run the previous command with the -i console option.

Note: To configure unattended, open a console window and then run the previous command with the "-i silent –f <configuration_properties_file>" options. The configuration properties file (ca-siebel-configuration.properties) is located in the install_config_info directory of the product.

Important! To configure unattended, you must configure the product using wizard or console once. The configuration properties file is required for unattended configuration. Before you perform the unattended configuration, verify that the configuration properties file is updated based on the host where you configure the product.

2. Provide the required information that was gathered before to complete the wizard.

Chapter 5: Post-Installation Configuration of Servers

This section contains the following topics:

Post-Installation Configuration for Policy Server

Create the Authentication Scheme

You can create the authentication scheme using the CA SSO Administrative UI.

Perform the following procedure to install the authentication scheme.

Follow these steps:

1. Navigate to the following location to access the Authentication Scheme library, SiebelSSOAuth:

   `<Siebel Agent Installation Directory>/siebel/bin`

2. Copy the library to the bin or lib directory of the Policy Server.
3. Open the Administrative UI and log in as an administrator.
4. Click Infrastructure, Authentication, Authentication Schemes.
5. In the Authentication Schemes page, click Create Authentication Scheme.
6. Specify the following values.

   Description

   Siebel SSO Agent Authentication Scheme

   (When you have completed the installation, you may optionally disable password acceptance by this Authentication Scheme.)

Authentication Scheme Type

Custom Template

Protection Level

5

In general, this value should be NO HIGHER than the value for any other HTML-based authentication scheme supporting username and password authentication.

Password Policies

Checkmark—if Password Services is in use

Library

SiebelSSOAuth

The Solaris and HP-UX platforms are case sensitive.

Secret

Enter a secret known only by the CA SSO administrator.

Confirm Secret

Re-enter the secret to confirm.

Parameter

A configuration string constructed from the following parameters separated by a semicolon:

- FCC=<URL for a login form>

  The FCC is the HTML page that collects credentials. This might be the URL that is used in another HTML-based authentication scheme. If you are unsure of the location of an appropriate FCC file on your system, and set the value for the FCC to:

  /siteminderagent/forms/login.fcc

  This displays the default CA SSO login form.

- ATTR=<User attribute>

  <User attribute> is the username that Siebel uses.

  Default: UID

- PERIOD=<Ticket Acceptance Period>

  The value, in seconds, for the maximum amount of time between the moment the SSO ticket is created and the moment a user might present a ticket. In general, this will be a very short period of approximately 10 to 20 seconds. The default is 60.

The order of the individual components in the parameter string is not important.

More information:

Disabling Password Acceptance (see page 41)

Create CA SSO Policies

CA SSO policies protect your Siebel applications and provide the framework for single sign-on (SSO). To create CA SSO policies to protect your Siebel resources, use the following process:

Follow these steps:

1. Open the Administrative UI.
2. Create a Policy Domain to contain all your Siebel applications, and do the following steps:
   a. Within the Policy Domain, create one realm for each Siebel application to be integrated. For example, create a /sales/ realm and a /purchase/ realm. M
   b. In each realm create two rules:
      - A rule with the actions Get and Post for the * resource
      - A rule with the Authentication event OnAuthAccept
   c. For each of the realms, make sure to select the Authentication Scheme that you created earlier.
   d. Create a response for SessionLinker. Siebel uses _sn as the name of the cookie. Thus, the correct configuration string for the SessionLinker is:

      "COOKIE=_sn"

   e. Create a policy with the following:
      - Add appropriate users.
      - Add both rules in each realm.
      - Link the SessionLinker response to the OnAuthAccept rule.

3. Create a Realm within the same Policy Domain for the startup URL (/SiebelConnector/).
   a. Make sure to select the Authentication Scheme you created earlier.
   b. In the CA SSO Agent Startup realm, create two rules:
      - One rule with the actions Get and Post for the * resource
      - One rule with the Authentication event OnAuthAccept
   c. Create a response (for example, Siebel connector response).
   d. In the CA SSO Response Attribute Editor, create two responses for the WebAgent-HTTP-Header-Variable attribute:
      - A response for the username. Specify the following values:

Attribute Kind	User Attribute
Variable Name	SIEBELUSER
Attribute Name	uid This value should be the attribute used by the directory to locate users (whatever user attribute contains the Siebel username, typically uid). To determine the correct value, examine the user directory configuration and the DN lookup start and end.

      - A response for the single sign-on (SSO) ticket. Specify the following values:

Attribute Kind	Active Response
Variable Name	Leave this blank
Library Name	SiebelSSOAuth
Function Name	GetSSO Ticket In some environments, the function name should be GETSSOTicketWithDN. See Upgrade and Enable the New Encryption Ticket.

Parameters	Enter a string constructed from the following values: ■ ATTR=<*User attribute*> <*User attribute*> is the username that Siebel uses. This is the value you entered in the Authentication Scheme's configuration string. ■ SECRET=<*Secret String*> This value should be same as the value entered in the Secret field of the Authentication Scheme. This parameter is mandatory. If you want to encrypt the secret, use the NPSEncrypt tool. See NPSEncrypt Tool. Parameters can be in any order, separated by semicolons.
Attribute Caching	Recalculate the value to a number that is less than the PERIOD setting in the Authentication Scheme, typically 60 seconds or less.

a. Click the Advanced tab and remove the equal sign (=), which, if it exists, is found before the less than (<) sign.

b. Create a Policy binding the OnAuthAccept rule with the response for all users that should have access to Siebel.

Post-Installation Configuration for Web Server

Configure the Startup Page

After installing the agent on web server, configure the startup page.

Follow these steps:

1. Locate the folder that is appropriate for the web server you are using, from the following list:
   - <*Siebel Agent installation folder*>\Siebel\Config\WWW\ASP (for Microsoft IIS)
   - <*Siebel Agent installation folder*>\Siebel\Config\WWW\CGI (for Apache-based web servers.)
   - <*Siebel Agent installation folder*>\Siebel\Config\WWW\JSP (for Sun ONE)

2. Copy the entire contents of the directory to a new folder named "SiebelConnector" within the web server's document root directory, for example:
   `C:\Inetpub\WWWRoot (for Microsoft IIS)`
3. For the Apache-based web server, do the following:
   a. Give Execute permissions to the perl (.pl) files in the SiebelConnector folder.
   b. Define the ExecCGI option for the SiebelConnector folder. Specify the cgi handler for the .pl files. The following sample configuration might help in defining in the httpd.conf file:

```
<Directory "/<Apache-based web server document root
directory>/SiebelConnector">
        AllowOverride None
        Options None EXecCGI
        Order allow,deny
        Allow from all
</Directory>
AddHandler cgi-script .pl
```

Do not change the files in any way. After the installation is operating properly, you may customize the files while preserving the functionality. Some of the files contain comments to help you customize the files.

Verify CA SSO Responses

To verify responses, open a web browser and access the relevant URL page, depending on your web server:

```
http://<machine.domain.com>/SiebelConnector/test.asp
http://<machine.domain.com>/SiebelConnector/test.jsp
http://<machine.domain.com>/SiebelConnector/test.pl
```

When loaded, this page verifies the contents of the CA SSO responses for the Siebel Username and Ticket.

- The CA SSO login page should appear. If the CA SSO login page does not appear, stop the installation and verify the CA SSO Policies.
- The Siebel Connector Test Page appears. Do one of the following:
  - If the page has green OK indicators and a message that no errors are found, verification is successful. Continue with the installation.
  - If the text has a red background and the description of a problem, verification is unsuccessful. Resolve the problem by following the suggestion on the test page. Rerun the test page after fixing the problem.

More information:

Post-Installation Configuration for Siebel Server (see page 35)
Create CA SSO Policies (see page 31)

Post-Installation Configuration for Siebel Server

Sample SmSiebelSSO.conf File

Following is a sample SmSiebelSSO.conf file.

```
# Feel free to edit this file at will

# Where should the log data go?
LogFile=c:/logs/Test.log

# Log level is an integer between 0 and 3
# Level       Meaning
#   0          None
#   1          Errors
#   2          Information
#   3          Debug
LogLevel=3
# These settings dictate how you communicate with the Policy server

AgentName=siebel1
HostConfigFile=/home/oracle/CA/erpconn/siebel/Config/SmHost.conf
Resource=/SiebelConnector/

# And finally how do you talk to the Database?
DatabaseUser=sadmin
DatabasePassword=sadmin
```

Executing the Security Adapter Test

After running the configuration wizard, test Security Adapter by verifying the installation and the settings in the configuration file (SmSiebelSSO.conf).

Follow these steps:

1. Run the ProviderTest75 program. The output from the execution is shown in the following code:

```
C:\>providertest
Enter username: dsherman
Enter password: password
Testing provider with username 'dsherman' and password 'password'
Loading library... OK
Finding entry point... OK
enter Config File: test.conf
Calling SecurityLogin()...OK
Return code is OK
Test 1: GetUsername()
        Username: dsherman
Test 2: GetAccountStatus()
        Account state: ACTIVE
Test 3: GetCredentials()
        pCred->m_pType:
        pCred->m_pUsername:  DBUser
        pCred->m_pPassword:  10 characters long
Test 4: GetUserInfo:
        m_accountStatus: ACTIVE
        m_bPasswordSet: 0
        m_pCredentialsArray
                #:       Type |   Username  |   Password
                ----------------------------------------
                0:            |     DBUser  |   10 chars
        m_pNewUsername: (null)
        m_p_Password: (null)
Test 5: GetRoles()
        GetRoles returned SecurityErrOK
        Role 00: A
        Role 01: B
        Role 02: C
        Role 03: D
        Total: 4 roles
```

2. If ProviderTest75 finds a problem, it displays messages such as Provider test failed or pUser is NULL, or something similar. Check the log file specified in the configuration file. If no log file is generated, check if the path to the configuration file is correct and that the user running ProviderTest75 has permission to open that file.
3. When the test is successful, enable the Security Adapter:

More information:

Enable Security Adapter (see page 37)

Enable Security Adapter

Perform the following steps to enable Security:

- Create named Subsystem for Custom Security Adapter (see page 37)
- Configure the Components to Use Custom Adapter (see page 39)

Create Named Subsystem for Custom Security Adapter

Perform the following procedure to create a named subsystem for Custom Security Adapter.

To create a named subsystem

1. Log in to the server manager using the following command:
   `srvrmgr` *`parameters`*

   The parameters include the following:

   - /e *ent_name*
   - /g *gtwy*
   - /s *srvr*
   - /u *username*
   - /p *password*
2. Create a named subsystem, SiteMinderSecAdpt, for the custom Security Adapter (sample output is shown in the following table), by using the following commands:

```
srvrmgr:srvr>create named subsystem SiteMinderSecAdpt for subsystem
InfraSecAdpt_CUSTOM
srvrmgr:srvr>list param for named subsystem SiteMinderSecAdpt
```

PA ALIAS	PA VALUE
CustomSecAdpt_CRC	********

CustomSecAdpt_SecAdptDLLName	
ConfigFileName	
CustomSecAdpt_HashAlgorithm	RSASHA1
CustomSecAdpt_HashDBPwd	False
CustomSecAdpt_HashUserPwd	False
CustomSecAdpt_PropagateChange	False
ConfigSectionName	
CustomSecAdpt_SingleSignOn	False
CustomSecAdpt_TrustToken	********
CustomSecAdpt_UseAdapterUsername	False

11 rows returned.

3. Modify the named subsystem created in Step 2 so that it uses the SiteMinder security provider library and configuration files, using the following commands:

   ```
   srvrmgr:srvr> change param CustomSecAdpt_SecAdptDllName=SmSecurityProvider75
   for named subsystem SiteMinderSecAdpt
   srvrmgr:srvr> change param ConfigFileName=d:\siebel\bin\enu\SmSiebelSSO.ini for
   named subsystem SiteMinderSecAdpt
   ```

   Note: The absolute path of the SmSiebelSSO.ini file must be specified in this command to modify the ConfigFileName.

   ```
   srvrmgr:srvr> change param ConfigSectionName=SiteMinder for named subsystem
   SiteMinderSecAdpt
   ```

   Note: The section name, SiteMinder, in this command to modify the ConfigSectionName, must match the section name defined in the SmSiebelSSO.ini file.

   ```
   srvrmgr:srvr> list param for named subsystem SiteMinderSecAdpt
   ```

   A sample output is shown in the following table:

PA ALIAS	PA VALUE
CustomSecAdpt_CRC	********
CustomSecAdpt_SecAdptDLLName	SmSecurityProvider75
ConfigFileName	d:\siebel\bin\enu\SmSiebelSSO.ini
CustomSecAdpt_HashAlgorithm	RSASHA1
CustomSecAdpt_HashDBPwd	False
CustomSecAdpt_HashUserPwd	False
CustomSecAdpt_PropagateChange	False

ConfigSectionName	SiteMinder
CustomSecAdpt_SingleSignOn	False
CustomSecAdpt_TrustToken	********
CustomSecAdpt_UseAdapterUsername	False

11 rows returned.

Configure the Components to Use Custom Adapter

Perform the following steps to configure the server components.

To configure the server components

1. Execute the following commands for the desired server component, such as esales_enu Object Manager:

```
srvrmgr:srvr > change param secadptname=SiteMinderSecAdpt for comp
eSalesObjMgr_enu
srvrmgr:srvr > change param secadptmode=CUSTOM for comp eSalesObjMgr_enu
```

2. Restart the Siebel Server.

Configure External Applications to Use SWELogin.swt

By default, customer-facing applications (such as eSales) use a different login view than the SWELogin.swt required by the CA SSO Agent. (Internal Seibel applications, such as CallCenter, use SWELogin.swt by default.) Perform the following procedure to configure other applications to use SWELogin.swt.

Follow these steps:

1. Open the eapps.cfg file present in *Siebel_SWSE_install*/bin folder.
2. Search the *application_language* file (such as esales_enu) for the string corresponding to the application for which the SWELogin.swt needs to be enabled and remove or comment the “startcommand” entry defined under it.
3. Restart the Web Server.
4. Open the application.cfg file (such as esales.cfg) from Siebel_Server_Root/bin/enu folder. Search for and comment the "LoginView = Login View" entry.
5. Restart the Siebel application server.

Test Security Adapter within Siebel

Perform the following procedure to test Security Adapter within Siebel.

Follow these steps:

1. Select a user whose password within Siebel is different from the CA SSO password.
2. After the server is restarted, open a web browser and access the selected application. For example, select the application esales by specifying:

   `http://machine.domain.com/esales/`

   Be sure to include the full domain name so that the browser accepts cookies of CA SSO. If a CA SSO session has not yet been established, the CA SSO login screen appears. Enter a valid username and password and complete the login process.
3. Verify the status of the operation, and perform the following:
   - If authentication is successful, the Siebel login screen appears. Provide the same credentials used to log into CA SSO and submit the form. Access to Siebel is granted. Close the web browser and open it again to destroy the existing CA SSO and Siebel sessions.
   - If authentication is unsuccessful, the Siebel login page *does not appear* and a Server Busy message is displayed. This could indicate that Security Adapter has not been installed, configured or activated correctly. Repeat the steps described in the following section:

     Execute the Security Adapter Test (see page 36).

Test Single Sign-On

When you verify that Security Adapter and Authentication Scheme are functioning correctly, SSO should also succeed.

Follow these steps:

1. Access the relevant URL, depending on your server:

   `http://machine.domain.com/SiebelConnector/testsso.asp`
   `http://machine.domain.com/SiebelConnector/testsso.jsp`
   `http://machine.domain.com/SiebelConnector/testsso.pl`
2. Enter the correct URL for the Siebel application (for example, esales), which you configured in Enable Security Adapter (see page 37).
3. Click Test Single Sign On. No additional login pages need to be presented and the application startup page automatically appears.

4. If single sign-on is successful, go to Direct Users Through the SSO Process (see page 41).
5. If single sign-on is unsuccessful, examine the Security Adapter log file and the relevant Policy Server log (either Authentication or Authorization) for additional information. The most common causes for failure are the following:
   - CA SSO responses.
   - Failing to select the Siebel SSO Auth Scheme for the realm used by Security Adapter. Re-check the Policy Server configuration.

Direct Users Through the SSO Process

Once single signon is functioning correctly, you can have all users automatically directed through the single signon process rather than presenting the Siebel login page.

To direct users through the SSO process

1. Locate the Siebel Login web template file:
   Siebel Agent Installation folder/Config/siebsrvr/Webtempl/SWELogin.swt
2. Copy it to the web templates (Webtempl) directory of the Siebel Server.
3. Modify the file to work with your web server (it is currently set up to work with .asp files). If you use .jsp or .pl files, open the SWELogin.swt file and change the instances of asp to jsp or pl respectively.

Additional Options

Disabling Password Acceptance

After completing the installation, you may select to disable the acceptance of passwords by the Authentication Scheme. Currently, the installation is configured in a way that allows a user, accessing the Siebel Object Manager, to provide a username and either a password or a single signon token generated through Active Response.

In many environments, Object Manager can be accessed through the Siebel Web Engine (SWE) components or via a thick client. Therefore, disabling password acceptance by the Authentication Scheme is valuable and necessary.

In other customer environments, once CA SSO is enabled, it mediates all access to the system; passwords are unacceptable and are considered a security risk. In these cases, access by a password can easily be disabled.

Once you disable the acceptance of passwords by the Authentication Scheme, the following occurs:

- Any realm protected by this Authentication Scheme will no longer accept a password, resulting in portions of the web site no longer accepting users.
- ProviderTest (or ProviderTest75) itself can fail, rendering troubleshooting extremely difficult.

Preventing Security Adapter from accepting passwords includes the following:

1. Make sure your environment is working properly.
2. Consider the implications and possibly create additional realms, rules and policies within CA SSO exclusively for use by Security Adapter. If you have any questions, contact CA for assistance.
3. Add the following text to the existing parameter for the SiebelSSOAuth Authentication Scheme:

   ```
   ;AcceptPassword=No
   ```

Providing Siebel Roles from CA SSO Policies

In addition to supporting single sign-on and authentication, this product has the ability to provide Siebel with a set of roles and responsibilities for individual users. The roles and responsibilities to be presented are collected from CA SSO responses by the connector at login time and are presented to the Siebel server whenever needed.

Note: The connector can add to a user's privileges but cannot remove roles and responsibilities configured within Siebel itself. This is an important consideration for privilege management because security can be compromised if roles and responsibilities are administered in both the enterprise directory and Siebel.

To provide roles to Siebel via the connector, create responses (and appropriate values) with the name SIEBELROLE. The connector does not attempt to validate the roles provided to Siebel; it simply passes to Siebel the values provided as responses for Siebel's use.

Using Load Balanced Web Servers with Siebel

This product does not impart any additional restrictions on load balancing. For information on configuring a web load balancer in a Siebel environment, see the Siebel documentation.

Security Adapter is a CA SSO Agent in its own right. Security Adapter is independent of the Web Agent and should not use the same agent name as any of the Web Agents in the environment.

When you configure policies, create an agent group, which should contain all of the Web Agents that will be protecting Siebel, and add the Security Adapter agents to that Agent Group.

To understand why each Security Adapter has its own agent name, consider the following environment as a similar case:

1. CA Access Gateway (formerly Secure Proxy Server) can be used as a front-end proxy to a number of web servers. Each web server can have a Web Agent installed. Each Web Agent is configured to use its own name; the fact that the user passed through CA Access Gateway makes no difference in the Web Agent configuration.
2. When this environment includes more than one CA Access Gateway in a load balanced configuration, the Web Agent configuration remains unchanged; it makes no difference to the Web Agents which CA Access Gateway instance sent the request to the web server, or even that CA Access Gateway was involved.

Using a number of web servers in front of a single Siebel Object Manager with this productl is virtually identical to CA Access Gateway and Web Agent environment described above. Access permissions to the Siebel Object Manager, protected by CA SSO Security Adapter, are predicated upon the policies and it makes no difference what web server was used to reach Object Manager.

Use a Different Authentication Scheme

You may use an authentication scheme other than Siebel SSO Authentication Scheme. For example, you might use SecurID or Certificates instead of the username and password-based authentication.

The following steps assume that the Web Agent and Siebel Security Adapter use different agent names and are in a common Agent Group. If both the Web Agent and Security Adapter are configured to use the same agent name, one of the agent names will need to be changed and the relevant system restarted.

To use a different authentication scheme

1. Open the SiteMinder Policy Management GUI and create another realm.
   a. For the agent, select the agent name used by Security Adapter.
   b. For the Authentication Scheme, select the Siebel SSO Auth scheme you already created.
   c. For the resource, enter /SecurityAdapter/
2. Create a rule for GET and POST to the resource *.
3. Create a Policy binding the GET/POST rule to the existing response for all users that should gain access to Siebel.

4. In the SmSiebelSSO.conf file, change the resource to:

   `/SecurityAdapter/`

5. Run ProviderTest (or ProviderTest75) to verify the new configuration.
6. Restart Siebel Object Manager.
7. Change the realm for the Web Agent to use the desired Authentication Scheme.
8. Remove the Security Adapter's realm from the Agent Group.
9. Retest the environment, paying particular attention to the log files.

Once the system is working properly, consider changing the Authentication Scheme's configuration to prevent it from accepting passwords.

More information:

Disabling Password Acceptance (see page 41)

Supporting Multiple Siebel User Attribute Responses for Siebel 7.8, 8.0.x, or 8.1.x

In the current design of the application, only one of the user attributes, such as uid, can be passed via the Siebel User attribute response header, SIEBELUSER.

The new parameter, UsernameHeaders, in the SmSiebelSSO.conf file, enables the SiteMinder agent for Siebel to support multiple Siebel User attribute response headers, as follows:

- In the SmSiebelSSO.conf file, set the UsernameHeaders parameter to multiple Siebel user response names in a comma-separated list, for example SIEBELUSER1,SIEBELUSER2, ...
- In the Policy server, create Siebel user attribute responses that are identical to the names entered as values in the UsernameHeaders A different user attribute can be configured for each Siebel user attribute response.

  Note: For the multiple user attribute functionality to be used correctly with the SiteMinder agent for Siebel, for any user request only one of the Siebel user attribute responses configured in Policy server should carry a value. All other Siebel user attribute responses should be empty for the signing in user.

- Based on your environment, modify the following Siebel agent web server files to check for the Siebel user response, which is carrying a value, and pass a valid non-null and non-empty value to the SWEUsername parameter in URL before redirection:
  - functions.inc, for ASP-based files.
  - getheaders.jsp, for JSP-based files.
  - siebelstartup.pl, for Perl-based files.

Note: If parameter UsernameHeaders is not configured in the SmSiebelSSO.conf file, the Siebel agent will continue to look only for the SIEBELUSER response.

Configure Policy Server Clusters

The CA SSO Agent for Siebel supports clusters of Policy Servers to improve performance.

Follow these steps:

1. Navigate to the following directory:

   <Siebel_Agent_Install_location>/siebel/bin

2. Locate the files for your operating environment from the following list:

   Windows

   Uses the following libraries:

   - smcommonutil.dll
   - smerrlog.dll

   Solaris and AIX

   Uses the following libraries:

   - libsmcommonutil.so
   - libsmerrlog.so

   HP-UX

   Uses the following libraries:

   - libsmcommonutil.sl
   - libsmerrlog.sl

3. Copy the libraries for your operating environment to the following directory:

   Siebel_ROOT/siebsrvr/bin

4. Open the following file with a text editor:

   SmSiebelSSO.conf

5. Add the following parameters (on separate lines) with the values you want:

   Clusters

   Specifies the IP addresses of several Policy Servers that are separated into smaller groups of clusters which help to improve performance. Separate individual Policy Servers in a cluster with spaces. Separate clusters with semicolons.

   Example: 127.0.0.100 127.0.0.101 127.0.0.102; 192.168.2.100 192.168.2.101 192.168.2.102;

   ClusterThreshold

Specifies the minimum percentage of available Policy Servers within a cluster. If the number of available Policy Servers within a cluster falls below this number, failover to the next cluster occurs. For example, if each cluster contains six Policy Servers, and the ClusterThreshold is 50, then failover occurs if *more than* three of those Policy Servers in the cluster are not available. When the value of this parameter is set to zero, *all* Policy Servers in a cluster must fail before failover occurs. When the value of this parameter is set to 100, failure of *one* Policy Server in a cluster triggers failover to the next cluster.

Default: 0

Limits: 0—100

6. Locate the line with the PolicyServer parameter, and add a comment character at the beginning.
7. Save and file and close the text editor.

   Policy Server clusters are configured.

   Note: If you decide not to use Policy Server clusters in the future, remove the comment character from the PolicyServer line. Add comment characters before the lines of the parameters in Step 2.

Chapter 6: Upgrading CA SSO Agent for Siebel

Use the CA SSO Platform Support Matrix available on the CA Support website to verify that both the previous and new versions of the product use the same operating environment. Upgrades are only supported if the operating environments for both versions are the same.

Upgrade the other CA SSO components (such as the Policy Server and Web Agent) in your environment first.

Note: For more information about upgrading the CA SSO components, see the CA SSO documentation.

Gather the information for the installer and run the installation wizard to install and configure the product.

This section contains the following topics:

Upgrade CA SSO Agent for Siebel on Windows

Important! Before you start the upgrade process, verify that the services of the web servers are stopped.

Follow these steps:

1. Double-click the following installation executable file:

   `ca-erp-siebel-<version>-<operating_environmentprocessor_type>.exe`

   For console-based installation, open a command line window and run the following command

   `<executable_file_name>.exe -i console`

2. Click Next.
3. Accept the License Agreement and click Next.
4. Select the components that are already installed on the host where you are installing the product.
   - Siebel application server
   - Web server, where the Web Agent has been installed.
   - Policy Server

   Note: Depending on your configuration, whether the above-mentioned servers are located on the same or on different machines, you must run Installer once, twice or three times.
5. Click Next.
6. Click Continue with the UPGRADE and click Next.

   You are prompted to overwrite the existing file.
7. Click Yes and click Next.
8. Review your selections, and click Install.

   After the installation is complete, you are prompted to configure the agent.
9. Select Yes, I want to configure now and click Next.
10. Complete the configuration wizard using the information gathered previously for the installer.

Upgrade CA SSO Agent for Siebel on UNIX

Important! Before you start the upgrade process, verify that the services of the web servers are stopped.

Follow these steps:

1. Run the executable installer file from the command line using the following command:

   `<executable file name> -i console`
2. After selecting the relevant servers that are installed on the machine that you are installing the agent, the installer prompts you to confirm the upgrade.
3. Enter Y.

   You are prompted to overwrite the existing file.
4. Enter Y and follow the instructions to complete the installation.

   If you are running the installer on the host that has the Siebel server installed, then run the configuration script to configure the agent.

5. Navigate to the following location:

   `<Siebel_agent_install_folder>/siebel`

6. Run the following script:

   `ca-siebel-config.sh`

7. Follow the instructions and complete the configuration.

Upgrade and Enable the New Encryption Ticket

Before upgrading to the latest release in a production environment, perform the installation in a test environment to make sure the upgrade does not have any adverse consequences.

Follow these steps:

1. Run smobjexport command to generate a backup of the existing Policy Store.

   Note: For more information, see the CA SSO documentation.

2. Stop one Policy Server.

3. Replace the existing file, SiebelSSOAuth.dll, libSiebelSSOAuth.so, or libSiebelSSOAuth.sl with the file in the following location:

   `<Siebel agent installation folder>\Siebel\bin`

4. Start the Policy Server.

5. Wait several minutes to allow all agents to begin again using this Policy Server.

6. Repeat Step 2 through Step 5 for each Policy Server in the environment.

7. Open the CA SSO Administrative UI and find the response for the Siebel Ticket. Change the Function Name from GetSSOTicket to GetSSOTicketWithDN (you do not have to flush the cache manually, it happens automatically).

8. Wait several minutes for every Policy server to detect the changed response and for the cache on every Web Agent to be flushed.

   Note: Manually flushing the cache does not accelerate this process.

9. Test the environment to ensure that Siebel is working correctly.

Chapter 7: Troubleshooting

This section contains the following topics:

Response Test or Session Startup Errors

Symptom:

An error occurred during the response test or on session startup.

Solution:

Verify the CA SSO Policies by using CA SSO Test Tool.

To verify CA SSO Policies

1. Click Start, Programs, CA, CA Single Sign-On, CA Single Sign-On Test Tool
2. Specify the correct Agent Name and SmHost.conf file.
3. Click Connect.
4. Enter the correct validation realm resource (for example, /esales/), the action GET, and click IsProtected.
5. Enter a valid CA SSO username and password. Click IsAuthenticated, and IsAuthorized.
6. If at any time a red indicator appears or if the NPS_SESSION_LINKER response does not appear in the Attributes box, examine the Policy Server configuration and logs. The logs are mandatory for proper configuration.
7. Change the resource to /SiebelConnector/. Click IsProtected, IsAuthenticated, and IsAuthorized. Verify that no red indicators appear and that responses appear for both SIEBELUSER and SIEBELTICKET.

Unable to Reach Siebel Startup or Siebel Login Page

500 Server Error

Symptom:

The web browser shows a 500 Server Error page or the web browser continuously returns to the CA SSO login page.

Solution:

Examine the Web Agent log.

Note: This problem does not relate to the Siebel SSO agent– it is a problem in the site's Web Agent configuration.

Server Busy Error

Symptom:

ProviderTest (or ProviderTest75) reported no problems but a message indicates that the server is busy or experiencing difficulties.

Solution:

Examine the Security Adapter logs.

Consider using the AnonUsername and AnonPassword settings.

Ticket Outside Acceptance Window Issue

Symptom:

The symptoms of this problem include an infinite loop in the browser window and the following message that appears in the Policy Server Authentication Log:

```
Ticket outside acceptance window - replay attack?
```

The most common problem encountered is an error in response creation, specifically in configuring attribute caching. Another probelm is or the time difference between the SIEBELTICKET generation and its validation by SiebelSSOAuth authentication scheme being higher than the configured PERIOD value in authentication scheme.

Solution:

To correct a ticket outside acceptance window issue, open the response in the CA SSO Administrative UI and adjust the Attribute Caching setting or increase the PERIOD parameter value in the authentication scheme.

More information:

Create CA SSO Policies (see page 31)

Agent API Not Loaded

Symptom:

Security Adapter attempts to dynamically load the CA SSO Agent API when needed. If the Agent API library cannot be found, the following message appears in the Security Adapter log file:

```
Agent API Not loaded
```

This message indicates that the system is unable to locate the relevant CA SSO Agent API file (SmAgentAPI.dll, libsmagentapi.so, libsmagentapi.sl).

Solution:

Check that the Agent API file is present.

If the file is present, but this error persists, do the following, according to your platform:

- AIX or Solaris: Do the following:
  - Update the LD_LIBRARY_PATH (or LIBPATH on AIX) to include the directory where the libsmagentapi.so is located
- HP-UX 11: Do the following:
  - Update the SHLIB_PATH to include the directory where the libsmagentapi.sl is located
- Windows: Copy the *Siebel agent installation folder*\siebel\bin\SmAgentAPI.dll file to the C:\WinNT\system32 directory.

Note: Within the Security Adapter file (SmSiebelSSO.conf), the settings for LogFile and LogLevel determine what information is logged. Make sure you have defined a log file and a level of logging.

Connecting to Server Error

Valid for CA SSO Agent for Siebel for HI client application

Symptom:

An error connecting to server message appears at the top of the Siebel application page when the CA SSO session times out before the Siebel session times out.

Solution:

Set the CA SSO session to a large value, and set the Siebel session timeout to a lower value so that Siebel governs the idle session timeouts.

Set the TurnLoopingOff variable as follows:

- 1 in the siebelstartup.asp file
- true in siebelstartup.jsp or siebelstartup.pl files.

Web Server Trace File Issue

Valid when Siebel WSE resides on IIS6

Symptom:

The web agent trace file on the web server does not log additional information.

Solution:

When Siebel WSE resides on IIS6, it creates a virtual folder for the Siebel application within the default website that has a different application associated to it.

Do the following:

Add the ISAPI6WebAgent.dll wildcard application mapping for the Siebel application/folder within the default website. This will cause the additional logging to appear in the webagent trace file.

Monitoring the Processing of a Request

The following stages in the processing of a request are documented in various log files:

- Generation of a Siebel authentication ticket (see page 57)
- Siebel User response (see page 57)
- Anonymous User Authentication (see page 57)
- Security Provider Contacts Policy Server (see page 58)
- Policy Server Contacts the Siebel SSO Authentication Scheme (see page 59)
- Security Provider Checks the SIEBELUSER Response (see page 60)

Generation of a Siebel Authentication Ticket

Generation of a Siebel authentication ticket is recorded in the Policy Server trace, as shown in the following example:

```
.
**************************************************
…….Siebel SSO Ticket Generation Parameters
**************************************************
.
.
Generating SSO ticket WITHOUT DN
.
[SIEBELTICKET=[NDSEnc-D]IhOoXn6KH6D9GMSQ2yQOywuZa4Hw+Qcr6zYdZ/oqzxM=]
```

Siebel User Response

Firing a Siebel user response, which sends a user attribute whose value maps to a valid Siebel user, is recorded in the Policy Server trace, as shown in the following example:

```
[SIEBELUSER=test]
```

Anonymous User Authentication

Anonymous user authentication is recorded in the Siebel Agent Security Provider logs, as shown in the following example:

```
.
Checking for Anonymous user
Anonymous user password correct
.
```

Security Provider Contacts Policy Server

The process in which Security Provider contacts Policy Server and accesses the protected resource /SiebelConnector/ can be seen in the Siebel Agent Security Adapter log, as shown in the following example:

```
.
.
SecurityLogin8() called
Username: 'test'
Password: *Not shown* (54 chars)
Config file already loaded
SmAgentConnection::Connect()
Checking for Anonymous user
Anonymous user, checking password
Invalid Anonymous password - user will be authenticated via SiteMinder
SecurityLogin8() calling AuthAzAndCollectResponse()
.
.
```

Policy Server Verifies the User Credentials

The process in which Policy Server uses the Siebel SSO authentication scheme to verify the user credentials can be seen in the Policy Server traces as shown in the following example:

```
.
.
[SiebelConnector: Authentication phase]
.
[SiebelConnector: Authenticating user with SSO ticket]
.
[SiebelConnector: Username to be validated is 'test']
.
[SiebelConnector: Validating token
[NDSEnc-D]LYwrQqKp9mugsmf6mdHid3MRaQch4iilKUzi+PD0oIw= for user test]
.
[SiebelConnector: Ticket decrypted to 19 bytes]
.
[SiebelConnector: Decrypted ticket - checking contents]
.
[SiebelConnector: Ticket parser results:]
.
[SiebelConnector: Time: 1132825779]
.
[SiebelConnector: LoginName: test]
.
[SiebelConnector: Ticket in acceptance window]
.
[SiebelConnector: Auth succeeded]
.
```

Security Provider Checks the SIEBELUSER Response

Security provider checks the SIEBELUSER response against the response that was extracted from the HTTP headers. This can be seen in Siebel Agent Security Adapter log, as shown in the following example:

```
.
AuthAzAndCollectResponse - Authentication ACCEPTED
AuthAzAndCollectResponse - Authorization ACCEPTED
Found SIEBELUSER Response
Usernames match
There are 0 responses saved
Credentials for user 'sadmin' accepted
User authenticated - returning SecurityErrOK
SecurityGetCredentials8() called
Requested credential type is ServerDataSrc
Returning SecurityErrOK
```

NTLM Authentication Fails

Symptom:

The ATTR attribute value that is set in the Active Response is ignored resulting in authentication failure while using the NTLM authentication scheme.

Solution:

Add the following parameter to the Active Response and set the value to Yes.

EnforceAttrUsage

Specifies that CA SSO Agent for Siebel does not ignore the value set in the ATTR attribute.

Values: Yes, No

Appendix A: NPSEncrypt and NPSVersion Tools

This section contains the following topics:

NPSEncrypt Tool

Sometimes, *secret* values must be stored in a configuration file. For security purposes, you may want to encrypt and store the encrypted form of these secret values. To do this, use the NPSEncrypt tool.

When a setting allows encrypted values to be used, this product will decrypt it before use. If the setting is not encrypted, the value entered will be used as is.

The NPSEncrypt utility takes plain text entered on the command line, encrypts it, and prints the result on the screen. The resulting encrypted text can be cut and pasted wherever it is needed.

A product that allows an encrypted value will automatically decrypt the value when needed.

Run the NPSEncrypt utility from the directory where it exists. The default location of the utility is as follows:

<Siebel_agent_install_folder>\tools\

To encrypt a value, use the command prompt and type the NPSEncrypt command followed by a space and followed by the text to be encrypted, as shown in the following example:

```
C:\>npsencrypt secret
[NPSEncrypt Version 1.1 - NPSEncrypt Revision 1]
[NDSEnc-B]CKtyevyWkrF24Aj9Ly+xEQ==
```

In this case the encrypted form of secret is as follows:

```
[NDSEnc-B]CKtyevyWkrF24Aj9Ly+xEQ==
```

When you copy and paste, grab the entire line, including the portion beginning with [NDSEnc-].

NPSEncrypt will encrypt the same text to many different cipher-text values. Use any of the values, for example:

```
C:\CA\siebel\tools\>npsencrypt secret
[NPSEncrypt Version 1.1 - NPSEncrypt Revision 1]
[NDSEnc-C]iQO2KVyRN2fB4tMwjtgRYQ==
C:\CA\siebel\tools\>npsencrypt secret
[NPSEncrypt Version 1.1 - NPSEncrypt Revision 1]
[NDSEnc-C]FWhVC+MiA7aNnA87szw76g==
C:\CA\siebel\tools\>npsencrypt secret
[NPSEncrypt Version 1.1 - NPSEncrypt Revision 1]
[NDSEnc-B]PD24A2Iz6H+KeDh7j4zUIg==
```

NPSVersion Tool

Use the NPSVersion tool to extract version information from many CA products. To use this tool, type the NPSVersion on a command line followed by a space and the name of the executable whose version information you want, for example:

```
C:\> NPSVersion sessionlinkd
    [NPSVersion Version 1.0 - NPSVersion Revision 1]
    sessionlinkd          -   Package: NPSSessionLinker V1.3
    sessionlinkd      - Component: SessionLinker daemon V1.3.2 (Jul 14 2003 20:26:16)
    sessionlinkd          -  Platform: AIX
C:\>
```

You may use the NPSVersion tool on one platform to extract information for a product built for any other platform. The actual information displayed might differ in format and content from that shown above, but the relevant lines when discussing any issues with Support are Package and Component. Each line has a version number.

Package refers to the version of the Product.

Component refers to the actual part of the product that is enclosed within this specific file. It is not uncommon for this version number to be larger than the *Package* version number. This is usually due to the Component having one of more bugs repaired or minor enhancements added, which did not require the entire Package to be rebuilt or renumbered.

Appendix B: Security Adapter Settings

The initialization and configuration files for Security Adapter are located in the *Siebel Agent Installation folder*:

<*Siebel Agent installation folder*>\Siebel\Config\SecurityAdapter\SmSiebelSSO.ini
<*Siebel Agent installation folder*>\Siebel\Config\SecurityAdapter\SmSiebelSSO.conf

You install these files on the Siebel Application Server's Object Manager, and modify the settings in the configuration file.

If you make any changes to the configuration file(s) after Security Adapter is enabled in Siebel, restart Object Manager.

This section contains the following topics:

Settings

LogFile

Specify the full path to the file that Security Adapter will use as a log file. On Windows systems, this path should include the drive, directory and file name. For example:

```
c:\logs\connector.log
```

On UNIX systems, the path should be absolute, for example:

```
/var/log/siebelconnector.log
```

LogLevel

Valid levels of logging are listed in the following table.

Level	Log Indicator	Meaning
0	*(not applicable)*	None; log file off.

Level	Log Indicator	Meaning
1	ERR	Errors only; errors in initialization and communication are logged.
2	INF	Informational; indicates the general cause of the problem.
3	DBG	Debug; information not typically useful in production environments.
4	XXX	Extra; helps locate problems in the Login Library code itself, and thus is typically not intended for non-CA personnel.

Note: At higher log levels (2 through 4) the file can increase in size very quickly. At a production site, using log level 1 is recommended.

PolicyServer

Security Adapter must communicate with the same Policy Server or servers as the Web Agent. You may specify a single Policy Server, multiple Policy Servers, or even one or more Policy Servers running on non-default ports.

Specifying a Single Policy Server

Minimally, you can set the IP address of a single Policy Server, for example:

```
127.0.0.1
```

In this case, Security Adapter assumes that the Policy Server is operating on the default ports.

Specifying Multiple Policy Servers

For multiple Policy Servers, you can specify all IP addresses or host names. You must enter them on a single line, separating them with a space. An example of two Policy Servers is:

```
192.168.1.4 192.168.1.5
```

In this case, Security Adapter assumes that the Policy Servers are operating on the default ports.

Specifying Ports

If you do not specify ports, Security Adapter assumes that the Policy Server(s) is/are operating on the default ports, which are the following:

1. 44441—Accounting port
2. 44442—Authentication port
3. 44443—Authorization port

If a Policy Server is not operating on the default ports, you must specify the ports. Use commas (not spaces) to separate the information items in the Policy Server string:

```
IPAddress,AccountingPort,AuthenticationPort,AuthorizationPort
```

For example:

```
192.168.1.4,44441,44442,44443
```

If multiple Policy Servers are using ports other than the default, you must separate each Policy Server string (which includes Policy Server and ports) by a space, for example:

```
192.168.1.4,44441,44442,44443 192.168.1.5,44441,44442,44443
```

Although the Accounting server might not be used and this product does not connect to the CA SSO Accounting server, 44441 is entered here for consistency with Web Agent configuration file syntax as well as to allow for future expandability. (You must specify the first port as the Accounting port, even though it is not being used internally.)

AgentName and HostConfigFile

Security Adapter uses AgentName and HostConfigFile settings to initiate connection to the Policy Server. AgentName must match the name entered in the CA SSO Administrative UI. The Host Configuration Object specified in the HostConfigFile must match the name specified in the CA SSO Administrative UI.

Action and Resource

The Action and Resource settings define the strings that the Security Adapter should send to the Policy Server when validating the user's ticket and their authorization to access Siebel. These strings are typically GET and /SiebelConnector/ and should only be changed by customers that fully understand CA SSO Policies and have a reason for not being able to use GET and /SiebelConnector/.

DatabaseUser and DatabasePassword

Once the Security Adapter is installed, Object Manager uses the username and password specified by the settings DatabaseUser and DatabasePassword whenever credentials are needed for communication to another system. The most common system for which Object Manager needs these credentials is the underlying database. Customers should refer to the section Select/Configure Database Credentials for additional information on the security implications of using the same credentials for all users when communicating with the database.

The setting DatabasePassword can be encrypted.

Credential Types

Using the same credentials to communicate with any other system is generally not a problem in test environments because administrative accounts tend to share a common set of credentials. In production environments, however, this can be a problem because security requirements typically dictate that common passwords may not be used for multiple systems.

To solve this problem, Security Adapter can be configured to return an alternative set of credentials for each credential type requested by the Object Manager.

For example, if a user has selected an administrative task and attempted to manage another Enterprise Server, the current Object Manager attempts to initiate communication with another Object Manager and fails. It displays a Login failed message.

In addition, the Server Manager's log file shows the following information:

```
(admauth.cpp 9(148) err=901042 sys=2) ADM-01042: Login failed for specified username, password, and data source
```

To configure Security Adapter properly, examine the Security Adapter's log file (with the log level set to 3) for the entry beginning with:

```
Requested credential type is ServerDataSrc.
```

To change the credentials returned, add two lines to the configuration file, one each for the username and password. Use the following format:

```
Credentials.Type.Username=<Username>
Credentials.Type.Password =<Password>
```

For example, the log file entries are:

```
Credentials.GatewayDataSrc.Username=sadmin
Credentials.GatewayDataSrc.Password=sadminpassword
```

The correct values vary, depending on your environment.

You may encrypt the credentials password by using the NPSEncrypt tool.

AnonUsername and AnonPassword

Once installed, Siebel Security Adapter is called by the Object Manager every time a username and password are presented. This feature allows CA SSO to integrate fully with Siebel and support both username and password-based signon, and the ticket-based single signon.

Having Object Manager call Security Adapter for every username and password presented does have one unintended consequence, which is that the Siebel Web Server Extension (WSE) connects to the Object Manager to download the Login page (typically SWELogin.swt file).

To make this connection, WSE sends a special username and password configured in the eapps.cfg file. By default, this username is SADMIN. When this username and password are sent to Security Adapter, Security Adapter passes it on to CA SSO for verification. As long as the user exists in the user store of CA SSO, with the password defined in eapps.cfg, WSE is able to download the login page. If the anonymous user does not exist in CA SSO, WSE returns an error saying that the server is either busy or experiencing difficulties. In these cases adding a special user to the user store is not a good solution.

This product allows sites to define one special user that this product will not verify against CA SSO. To ensure security, use this feature only for the WSE.

The configuration settings AnonUsername and AnonPassword can be set to the username and password specified in the eapps.cfg file. These are case sensitive; sadmin is not the same as SADMIN. This is intended to match the behavior of most user directories supported by CA SSO.

To encrypt AnonPassword, use NPSEncrypt.