CA Harvest Software Change Manager

Implementation Guide

Release 12.5

CA Technologies Product References

This documentation set references the following CA Technologies products:

- CA Harvest Software Change Manager (CA Harvest SCM)
- CA Software Delivery
- CA IT Client Automation (formerly, CA Desktop and Server Management, CA IT Client Manager)
- CA Clarity Agile
- CA Clarity Requirements

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: Introduction

This section contains the following topics:

Overview

This guide provides you with the information necessary for a successful CA Harvest SCM implementation in your enterprise. This guide provides information about how to do the following:

- Plan and prepare for both a *new* installation and an upgrade.
- Install and upgrade all of the necessary product components.
- Install, configure, and maintain the database on which the product runs for best performance.
- Configure the product components (such as the broker, server, and agents), authentication, integrated products, security, and additional product features.
- Troubleshoot certain circumstances that you may encounter.

Audience

This guide is intended for anyone who wants to understand how to install, upgrade, and configure CA Harvest SCM. The following users may have specific roles or combined roles during the implementation phase:

- *System administrators* use the information in this guide, and their operating system knowledge, to install the product for the first time, upgrade the product from release to release, and configure the product based on your implementation requirements.
- *Database administrators* use the information in this guide to install the database on which CA Harvest SCM runs, and configure the database for best performance.
- *Users* can use the information in this guide to install the client and agent, if necessary.

How to Implement the Product

The starting point and steps you use to implement CA Harvest SCM is based on a number of factors, such as whether this is the first time you are installing it, you are upgrading from a previous version, the operating system you have, the database on which the product runs, and so forth. In general, you should use the following steps for a successful implementation:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not start your implementation until you have read and understood that information.

   Note: You can find the most current version of the Release Notes at http://ca.com/support http://www.ca.com/us/support.aspx.

2. Verify that you have read, and have a basic understanding of, the product components.

3. Verify that you have read the implementation considerations.

4. Verify that you have your installation media (DVD).

   Important! If your computer does not have a DVD drive, contact Technical Support at http://ca.com/support http://www.ca.com/us/support.aspx.

5. If you are performing a *new installation* (that is, this is the first time you are installing the product), follow the platform-specific steps to install the server, client, and agent in the appropriate installation chapter.

6. If you are *upgrading* from a previous release, follow the platform-specific steps to upgrade the server, client, and agent in the appropriate upgrade chapter.

   Important! An upgrade on any platform means that you install the CA Harvest SCM Release 12.5 server software and then upgrade the database. We recommend that you back up the database before upgrading.

7. After you have completed your new installation or product upgrade, configure your implementation using the information in the chapter "Configuring the Product."

8. If you have any custom components (such as SDK components), set up and test the components using the information in the chapter "Customizing the Product."

More information:

Components (see page 17)
Implementation Considerations (see page 18)

Components

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

CA Harvest SCM includes several components that work together in a client/server environment running Microsoft Windows, UNIX, Linux, and z/OS. Specific platform support is discussed in the Release Notes. Before you begin your implementation, you should have a basic understanding of the following CA Harvest SCM components:

Server

The server program executes operations requested by CA Harvest SCM clients. Install this program on your application server.

Client

The client program manages the user interface on each user's computer by receiving input from the user and sending the information to the server for processing, and presenting information returned from the server to the user. Depending on the platform and installation options, the client may include the agent, administrator, command-line utilities, HSDK, JHSDK, the Web Interface, and the Workbench.

Workbench

The Workbench is a graphical user interface (GUI) application that provides access to all user functions of the product. The Workbench is supported on Windows, Mac, and Linux and is usually installed on users' computers.

Database

The product maintains the data under its control in a local or remote database, and the server and database use Java database connectivity (JDBC) to communicate.

Communication Services

The communication services consist of a set of protocols that enable the product components to work together and provide the connection between the server and client.

Broker

The broker program is a communication service. Because many clients and servers exist in one network, the broker assigns each client an appropriate server. Each server registers with the broker and provides the broker with the information necessary for the client to find the appropriate server when requesting service. Each client then requests the broker to connect to an available server.

Command-Line Utilities

The command-line utilities enable CA Harvest SCM clients to execute scripts and common processes from a UNIX or Linux shell or Windows command prompt. On all supported operating systems, the command-line utilities are a client.

Web Interface

The Web Interface (formerly known as Harweb) lets users access the product through a web browser. The Web Interface provides access to both administrative and user tasks, reducing the need to install the client and administrative applications locally.

HSDK

The product includes a Software Development Kit (HSDK), which consists of a set of C++ classes that let you manage and edit the product data objects and provides a set of platform-independent client-side interfaces to access the data.

JHSDK

The Java SDK for the product (JHSDK) is a set of Java classes that act as an interface between Java and the HSDK. The JHSDK helps ensure that the Java programs you execute with the HSDK are compatible with the product and comply with its standards.

Implementation Considerations

Before you implement CA Harvest SCM, there are certain choices and decisions that you should consider to help plan for a successful implementation. Consider the following:

- The Windows Upgrade Wizard (see page 19)
- The Database (see page 19)
- Lifecycle Project Templates (see page 19)
- Workbench Form Templates (see page 19)
- Client Access (see page 20)
- The Client Installation Method (see page 20)
- Remote Computer Access (see page 20)
- The Server Behind a Firewall (see page 20)
- CA Software Delivery Integration (see page 20)
- External and Internal Authentication (see page 21)
- Account Security Options (see page 21)
- Web Interface (see page 21)

- Custom Forms (see page 22)
- Third-Party Product Integration (see page 22)
- CA Harvest SCM Reports (see page 22)

The Windows Upgrade Wizard

If you use the Windows Upgrade Wizard to upgrade an existing product release such as Release 12.x for the Client and Agent components to the current release, you do not need to perform any additional tasks. For the Server component you need to run the HDBsetup command to upgrade the database.

The Database

Consider the following information when you select a database:

- You must install your database *before* you begin using CA Harvest SCM.
- If you select Microsoft SQL Server (SQL Server), both the server and SQL Server must run on Windows.

  Note: For information about installing your database, see your vendor documentation.
- The product database can run on the same computer as the server (*local* database) or on a different computer (*remote* database). Local databases are easier to install, configure, and maintain.

Lifecycle Project Templates

Lifecycle project templates are *not* loaded by the base database configuration scripts. Instead, the templates are loaded separately as an option of the hdbsetup Database Configuration Utility from CA Harvest SCM archive files (for example, a .har file).

Note: For information about the hdbsetup Database Configuration Utility, see The hdbsetup Database Configuration Utility (see page 223).

Workbench Form Templates

The Workbench uses XML form template definitions that are stored in the product database to retrieve form definitions. The form definitions are loaded separately as an option of the hdbsetup Database Configuration Utility (see page 223).

Client Access

The CA Harvest SCM client manages the user interface on each end-user computer. Every user needs access to a locally installed client, a client installed on a network, or the Web Interface.

The Client Installation Method

To install CA Harvest SCM clients on end-user computers, you can use any of the following installation methods:

- Let users copy and install the client locally, choosing their own installation options.
- Install the client locally for users and set the installation options for them.
- Install a client on a network, so that users can access the client from their local computers and run it from the network (Windows only).

Remote Computer Access

CA Harvest SCM agents act as file servers on remote computers, enabling users to check in and check out files stored on the remote computer. If users need access to a remote computer, install the agent on that computer.

The Server Behind a Firewall

If you plan to run the CA Harvest SCM server on Windows behind a firewall, we strongly recommend that when you install the server, you specify a *range* of available ports for the server to use.

More information:

Install the Server (Custom Installation) (see page 32)

CA Software Delivery Integration

When you install the CA Harvest SCM server, you can optionally install the CA Software Delivery (previously named Unicenter Software Delivery) integration to deploy software. You can use the product and CA Software Delivery together to deploy software.

Note: For more information about the platforms that the CA Software Delivery agent supports, see the CA Software Delivery documentation.

More information:

CA Software Delivery Integration Configuration (see page 321)

External and Internal Authentication

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

To authenticate CA Harvest SCM users' logon credentials for accessing remote computers, you can use external authentication (such as OpenLDAP authentication), internal application authentication, or a combination of external or internally authenticated user accounts. OpenLDAP authentication is an installation option for the product servers and agents running on Windows, UNIX, and Linux. If you plan to use OpenLDAP authentication, a supported OpenLDAP server is required. The product also considers internally and externally defined user groups. Mixed-mode authentication is an option when installing external authentication, to allow specifying which user accounts are to be authenticated internally.

Account Security Options

CA Harvest SCM provides a user account option, *Single Workstation Login*, to prohibit a second client session from being created unless the client in running on the same workstation. Your administrator must configure individual user accounts to use this option.

Note: For information about creating a user and configuring accounts, see the *Administrator Guide*.

Your server environment must also be configured so that the Web Interface does not include specific server computers on the list of computers following this restriction.

The Web Interface

The Web Interface is the web-based interface for the product, letting users access the product using a web browser. You can optionally use the Web Interface instead of using a local or network client. The Web Interface requires that the CA Harvest SCM command-line utilities be installed locally, and that the Java SDK Standard Edition be installed on your application server. The product provides an easy-to-use wizard for upgrading or installing the Web Interface.

Custom Forms

For new installations in which you want to use custom forms to track change information, you must create the forms and make them available to clients. If you used custom forms from your previous CA Harvest SCM installation, convert them for use in the current release of the product.

More information:

How to Convert Customized Form Types and Add Them to the Database (see page 211)

Integrations

CA Harvest SCM supports the following integration tools:

Note: For the latest certifications of platforms and third-party tools used with CA solutions, contact Technical Support at http://ca.com/support.

- CA Harvest SCM Plug-In for Microsoft Visual Studio—Lets the product integrate into development environments using the Microsoft Visual Studio Integration package.

  Note: For more information about integrations that use Microsoft Visual Studio, see the *Plug-In for Microsoft Visual Studio User Guide*. You install and maintain this plug-in separately from the product components.

- CA Harvest SCM Plug-in for Eclipse—Lets the product access the product lifecycle using the Eclipse Developer environment.

  Note: You install and maintain this plug-in separately from the product components. For information about this plug-in, see the *Plug-In for Eclipse User Guide and* the CA Harvest SCM Plug-in for Eclipse web site https://support.ca.com/irj/portal/anonymous/phpsupcontent?contentID={77600E96-96DA-4D04-A9EA-EDBC7CF2FE38}.

CA Harvest SCM Reports

CA Harvest Software Change Manager (CA Harvest SCM) uses BusinessObjects InfoView as a business intelligence (BI) portal to collect, consolidate, and present your organization's CA Harvest SCM data. Reports include existing Dashboard, CA Harvest SCM reports, and new reports that cover security, audit, project change activity, package change activity, and source item change activity. The reports are useful for administrators, managers, quality assurance testers, and developers.

More information:

How to Install CA Harvest SCM Reports (see page 179)

Chapter 2: Installing on Windows

Important! Installing these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

How to Prepare for the Server Installation

The current release of CA Harvest SCM is packaged with CAPKI (latest version of ETPKI).

To help ensure that you successfully install the CA Harvest SCM server, complete the following steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the server until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Decide whether to perform a *typical* installation or a *custom* installation.
   - Select the *typical* option to:
     - Install the product using predefined settings without providing any additional input.
     - Install the server, the command-line utilities, and product documentation.
     - Install these common, shared CA components: Enterprise Communicator (PEC), eTPKI, and CA Licensing.
     - Automatically use internal authentication for product users.
   - Select the *custom* option to:
     - Configure settings in each step of the installation process.
     - Decide which components to install, including the server, command-line utilities, and product documentation.
     - Install these common, shared CA components: Enterprise Communicator (PEC), eTPKI, and CA Licensing.
     - Decide to use the Windows service for the product broker.
     - Install CA Software Delivery.
     - Enable FIPS mode for the product agent.
     - Specify a firewall port range.
     - Decide whether to use internal, OpenLDAP, or Mixed Mode Authentication for product users.
3. Determine the home directory (%CA_SCM_HOME%) in which you want to install the server. The default home directory for 32-bit Windows is C:\Program Files\CA\SCM. The default home directory for 64-bit Windows is C:\ Program Files (x86)\CA\SCM.
4. Determine whether to install CA Software Delivery.
5. Select an authentication method, either internal authentication (CA Harvest SCM), OpenLDAP, or Mixed Mode Authentication.
6. If you plan to use *Oracle* as your product database, consider the following information:
   - You must install Oracle before using any product component, including the Database Configuration Utility. You must run this utility to set up your product database on Oracle before you can use the product to create users or to check in and check out files. We recommend that you install Oracle before installing the product, so that you can configure your product database immediately after installing the product server.
   - If the Oracle database and product server are installed on different computers, verify that the Oracle client is installed on the product server computer.
   - Use the Database Configuration Utility to set up the CA Harvest SCM database.

- If you plan to use a *remote* Oracle database, verify that the following conditions are met:
  - The Oracle client networking utilities are installed.
  - You understand how a product server using a remote Oracle database connects to a product database using Oracle client TCP/IP utilities.
  - A version of Oracle supported by the product is installed on the remote computer.

  Note: For information, see your Oracle documentation and the Release Notes.

7. If you plan to use *SQL Server* as your product database, consider the following information:
   - You must install SQL Server before using any product component, including the Database Configuration Utility. You must run this utility to set up your product database on SQL Server before you can use the product to create users or to check in and check out files. We recommend that you install SQL Server before installing the product, so that you can configure your product database immediately after installing the product server.
   - When you install SQL Server, do *not* install it as case-sensitive unless you have a compelling reason to do so.
   - To verify the integrity of the disk I/O subsystem, run the SQLIOSTRESS utility on new systems before installing the product.
   - If you want to create and configure the CA Harvest SCM database, verify that your Windows operating system user ID is granted the *system* administrator role in SQL Server. If you want to create and configure the product database immediately after installing the product server, verify that you grant this role *before* you begin the product server installation.
   - Decide whether to use Windows authentication or SQL Server authentication for the product database connection. Before you decide, verify the authentication method that your server is using:

     Important! If you plan to use Web Interface, you *must* use SQL Server authentication mode.
     - If the server is set to Windows authentication, then you *must* set your product database connection to use Windows authentication.
     - If the server is set to Windows and SQL Server authentication (mixed mode), then you can set your product database connection to use either Windows authentication or SQL Server authentication.

     Note: For information about how to check and change the authentication mode, and grant the system administrator role, see your SQL Server documentation.
   - If the SQL Server database and the CA Harvest SCM server are installed on different computers, verify that the SQL Server client is installed on the product server computer.

More information:

CA Software Delivery (see page 28)
Authentication (see page 29)

CA Software Delivery

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

If you want to use both CA Harvest SCM and CA Software Delivery in your environment, you can optionally install CA Software Delivery during a custom installation of the product server. When you install CA Software Delivery, you can use special life cycle processes in the product to deploy software from CA Software Delivery. This feature extends the lifecycle support provided by the product by adding the ability to deliver updated software. You can optionally deploy to different targets at different times. For example, you can deploy to a small development testing team first, then to a quality assurance (QA) team, and finally to groups of users.

Using the predefined lifecycle named Deploy Release Model, or a customized lifecycle similar to it, you can stage, create, and deploy CA Software Delivery packages for distribution to target computers and computer groups. You select whether to stage, create, and deploy the software in three separate processes, or in a single step. Furthermore, you can optionally stage and create the packages yourself and leave the final process (deploying the software) to the CA Software Delivery administrator.

Important! To use this feature, you must install the optional web services support when you install the CA Software Delivery server. For instructions, see your CA Software Delivery documentation. The CA Software Delivery server that the product supports runs only on Windows Server and Linux.

However, because of the cross-platform support that both the product and CA Software Delivery provide, you can use them together to deploy software on *any* platform that the CA Software Delivery agent supports. For details about these platforms, see your CA Software Delivery documentation.

Note: For more information about using the product and CA Software Delivery together to deploy software, including details about using the predefined Deploy Release Model lifecycle, see How to Enable the CA Software Delivery Integration (see page 321).

Authentication

During the CA Harvest SCM installation, you can select to use either *internal* authentication, *OpenLDAP* authentication, or *Mixed Mode Authentication* as the method your site will use to authenticate users' names and passwords. The authentication is used, for example, when a user attempts to log in to the product.

- Internal authentication uses the product to authenticate the user name and password.
- OpenLDAP authentication uses an OpenLDAP authentication server for authentication.
- Mixed Mode authentication lets the SCMAdmin create users internally even though the authentication mode may be set to External (LDAP).

  Note: Mixed Mode authentication does not use LDAPserver for Authentication when users are created internally.

The authentication method that you select may depend on your company's IT standards and conventions, resources, environment-specific concerns, manager input, in addition to other factors.

If you select internal authentication, you do not need to perform any preparation tasks.

How to Prepare for the OpenLDAP Installation

Based on your planned security-related implementation, you may decide to use OpenLDAP authentication instead of internal authentication.

Follow these steps:

1. Verify that your LDAP server is installed and configured.

   Note: For a list of supported LDAP servers, read the Release Notes.

2. Decide whether to use Transport Level Security (TLS), Secure Socket Layer (SSL), or no security to encrypt communication between the product and your LDAP server. Specify TLS *only* if your LDAP server supports StartTLS.

   Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

3. If you select TLS or SSL, determine and record complete path names for the following:
   - The TLS trusted certificate (optional)
   - The TLS client certificate (optional)
   - The TLS client key (optional)
4. If you select TLS or SSL, decide which method you will use to supply the TLS values when you are asked to supply them during the installation:
   - By entering the actual path names.
   - By entering the name of the OpenLDAP configuration file that specifies these path names.

More information:

External Authentication Configuration (see page 297)

Install the Server (Typical Installation)

Install the CA Harvest SCM server and select the *typical* installation type to:

- Install the product using predefined settings without providing any additional input.
- Install the server, the command-line utilities, and product documentation.
- Install these common, shared CA components: Enterprise Communicator (PEC), eTPKI, and CA Licensing.
- Automatically use internal authentication for product users.

Note: If you want to install the server, determine the components to install (including external authentication such as OpenLDAP and the CA Software Delivery integration), and use a custom installation instead.

To install the server

1. Insert the installation media into your drive.

   The Product Explorer dialog appears.

2. Select the product component that you want to install.

   Note: If your computer has a previous release of the product installed, you are prompted to remove it. You must uninstall the previous release before you can install the new release.

3. To continue with the installation, follow the on-screen instructions.

4. When prompted, click Next to accept the default installation directory or click Change to select a different location.

   Important! If another product component, such as the client, is already installed on this computer, you *must* install the server in the same location as that component.

5. If you change the default installation directory, the installation directory you specify will be the top-level directory for the product files. For example, %CA_SCM_HOME% will be defined as *your-installation-path*\SCM. If the directory you specify does not exist, it will be created.

   Note: If you specify a nondefault installation path that includes an ampersand (&) in a folder name, the ampersand is included in the path name during installation. However, the ampersand does not appear in the path name in the destination folder during installation. Instead, the ampersand is removed and the next character after it is an underline. For example, if the folder name is CA&SCM, CA_SCM is displayed. However, the folder name used for the installation is CA&SCM.

   The Setup Type window appears.

6. Select Typical as the installation type and click Next.

7. Continue following the on-screen instructions to complete the typical installation.

   When the installation is complete, you can start the Database Configuration Utility to configure the product's database.

   Important! If you are using Oracle or SQL Server, you can configure your database now or at another time. However, before you begin using the product, you must configure your database.

More information:

Install the Server (Custom Installation) (see page 32)
How to Configure the Repository Using the hdbsetup Database Configuration Utility (see page 235)

Install the Server (Custom Installation)

Install the CA Harvest SCM server and select the *custom* installation type to:

- Configure settings in each step of the installation process.
- Decide which components to install, including the server, command-line utilities, and product documentation.
- Install these common, shared CA components: Enterprise Communicator (PEC), eTPKI, and CA Licensing
- Decide to use the Windows service for the product broker.
- Install CA Software Delivery.
- Enable Federal Information Processing Standard (FIPS) mode for the product agent.
- Specify a firewall port range.
- Decide whether to use internal, OpenLDAP, or Mixed Mode authentication for product users.

Note: If you want to install the server using predefined, default settings, you should use a typical installation instead.

Follow these steps:

1. Insert the installation media into your drive.
2. The Product Explorer dialog appears.
3. Select the product component that you want to install.

   Note: If your computer has a previous release of the product installed, you are prompted to remove it. You must uninstall the previous release before you can install the new release.
4. To continue with the installation, follow the on-screen instructions.
5. When prompted, click Next to accept the default installation location or click Change to select a different location.

   Important! If another product component, such as the client, is already installed on this computer, you *must* install the server in the same location as that component.

6. If you change the default installation directory, the installation directory you specify will be the top-level directory for the product files. For example, %CA_SCM_HOME% will be defined as *your-installation-path*\SCM. If the directory you specify does not exist, it will be created.

   Note: If you specify a nondefault installation path that includes an ampersand (&) in a folder name, the ampersand is included in the path name during installation. However, the ampersand does not appear in the path name in the destination folder during installation. Instead, the ampersand is removed and the next character after it is an underline. For example, if the folder name is CA&SCM, CA_SCM is displayed. However, the folder name used for the installation is CA&SCM.

   The Setup Type window appears.

7. Select Custom as the installation type and click Next.

   The Custom Setup dialog appears.

8. Select Server (the product server), any additional components you want to install, and click Next.

9. Specify whether to use the Windows service for the product broker by selecting the associated check box.

10. Specify whether to install the CA Software Delivery Integration and click Next.

11. If you decide to install the integration, configure the integration on the CA Software Delivery Integration Window (see page 34) and click Next.

12. Specify the authentication method to verify the login credentials for users and click Next.

    Internal

    Specifies internal (CA Harvest SCM) authentication. Login credentials provided to the broker are validated against the internal product user data.

    OpenLDAP

    Specifies an external server. Login credentials provided to the broker are validated against the external authentication server. If you select this authentication method, configure the LDAP settings on the LDAP Compliant Directory Configuration Windows (see page 62).

    Mixed Mode

    Specifies to use both internal (CA Harvest SCM) and external (OpenLDAP) authentication. The login credentials provided to the broker are validated against the external authentication server and the product if the user is external; otherwise, the login credentials provided to the broker are validated against the internal product user data if the user is internal. If you select this authentication method, configure the LDAP settings on the LDAP Compliant Directory Configuration Windows (see page 62).

    The Firewall Port Range dialog appears.

13. (Optional) Specify a range of ports for the server and click Next. If you do not specify ports, the server uses the Windows ephemeral (random) port range.

    Important! If you plan to run the server behind a firewall, we strongly recommend that you specify a *range* of available ports for the server to use. When you specify your ports, note that the port range is inclusive, the number of ports in the range *must* be greater than or equal to the maximum number of server processes and remote agents running behind the firewall, and this port range does *not* include the RTserver port (its default value is 5101).

14. Continue following the on-screen instructions to complete the custom installation.

    When the installation is complete, you can start the Database Configuration Utility to configure the product's database.

    Important! If you are using Oracle or SQL Server, you can configure your database now or at another time. However, before you begin using the product, you must configure your database.

More information:

Install the Server (Typical Installation) (see page 30)
How to Configure the Repository Using the hdbsetup Database Configuration Utility (see page 235)

The CA Software Delivery Window

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

If you install CA Software Delivery (CA SDM), set the following parameters to configure the communication and the synchronization of data between CA Harvest SCM and CA SDM.

USD Server URL

Defines the network (IP) address or host name of the CA Harvest SCM server on which CA Harvest SCM will stage, create, and deploy CA SDM packages.

For a CA SDM server running on *Windows*, use the following format for the URL:

http://*server hostname or network address*/UDSM_R11_WebService/mod_gsoap.dll

For a CA SDM server running on *Linux*, use the following format for the URL:

http://*server hostname or network address*/UDSM_R11_WebService

Default: None

Limits: 255 characters

Note: The USD Server URL is stored in the usdserver=parameter in the \CA_SCM_HOME\HServer.arg file. For more information about viewing and updating the parameters in this file, see Set the URL for the CA Software Delivery Server (see page 322).

Example: Specify a Server URL Using the Server Name (Windows)

In this example, to specify the URL for a CA SDM server on Windows, using the server's name (usdsrv01), enter the following parameter:

http://usdsrv01/UDSM_R11_WebService/mod_gsoap.dll

Example: Specify a Server URL for a Network Address (Linux)

In this example, to specify the URL for a Linux server whose network address is 138.42.44.57, enter the following parameter:

http://138.42.44.57/UDSM_R11_WebService

Note: For more information about this URL format, see the information about the login web service in the *Unicenter Desktop and Server Management (DSM)* documentation.

USD Server User Name

Defines the user name for CA Harvest SCM to use to log on to the USD server.

Note: CA SDM is not required to be installed before the CA Harvest SCM server is installed. Therefore, the logon credentials for the USD server and CA Harvest SCM remote agent on the USD server are not validated at installation time. Instead, they are validated when you attempt to invoke one or more “USD” processes from CA Harvest SCM or to synchronize the CA SDM and CA Harvest SCM databases.

For a CA SDM server running on *Windows*, use the following format for the user name:

winnt://*login domain or host name*/*username*

For a CA SDM server running on Linux, use the following format for the user name:

unixl://*login domain or host name*/*username*

Default: None

Limits: 255 characters

Limits: 255 alphanumeric characters

Important! CA Harvest SCM automatically stores the USD server user name and password (if specified) in the \CA_SCM_HOME\husdsr.dfo file and encrypts this file, using the svrenc utility. To update the USD Server user name, password, or both, you must use the svrenc utility. For more information about using this utility to update them, see the *Command Line Reference Guide*.

Example: Specify the User Name Using the User's Domain (Windows)

In this example, to specify the user name (usdadmin) for a CA SDM server on Windows, using the name of the user's domain (domain01), enter the following parameter:

winnt://domain01/usdadmin

Example: Specify the User Name Using the Server Host Name (Linux)

In this example, to specify the user name (usdadmin) for a CA SDM server on Linux, using the host name (usdsrv01) of the CA SDM server, enter the following parameter:

unixl://usdsrv01/usdadmin

Note: For more information about this user format, see the information about the login web service in the *Unicenter Desktop and Server Management (DSM)* documentation.

USD Server Password

(Optional) Defines the password of the user specified in the USD Server User Name field. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

CA Harvest SCM Agent User Name

Defines the user name for CA Harvest SCM to use to access the CA Harvest SCM remote agent running on the USD server.

Note: CA SDM is not required to be installed before the CA Harvest SCM server is installed. Therefore, the logon credentials for the USD server and CA Harvest SCM remote agent on the USD server are not validated at installation time. Instead, they are validated when you attempt to invoke one or more "USD" processes from CA Harvest SCM or synchronize the CA SDM and CA Harvest SCM databases.

Default: None

Limits: 255 characters

Important! The product automatically stores the USD CA Harvest SCM agent user name and password (if specified) in the \CA_SCM_HOME\husdra.dfo file and encrypts this file, using the svrenc utility. To update the USD CA Harvest SCM Agent user name, password, or both, you must use the svrenc utility. For more information about using this utility, see the *Command Line Reference Guide*.

CA Harvest SCM Agent Password

(Optional) Defines the password of the user specified in the CA Harvest SCM Agent User Name field. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

CA Harvest SCM Agent Port

Defines the port for running the CA Harvest SCM agent.

Default: None

Minimum: 2

Maximum: 9999

Note: The agent does not run on port 1 or 1000.

Synchronization Interval Between SCM and USD Server

Defines the interval (in minutes) at which CA Harvest SCM synchronizes certain tables in the CA Harvest SCM database with those in the CA SDM database. CA Harvest SCM checks the contents of these tables in the CA SDM database and, if necessary, updates the CA Harvest SCM tables to match the CA SDM tables.

At this synchronization interval, CA Harvest SCM also queries for the status of any outstanding deployment jobs it has scheduled. For those jobs whose status has changed, the HARUSDPLATFORMINFO and HARUSDHISTORY tables are updated.

For more information about these CA Harvest SCM tables and others, contact Technical Support at http://ca.com/support.

Default: 60

Minimum: 15

Important! If you do not set this parameter, no synchronization occurs between CA Harvest SCM and the CA SDM server.

Limits: 6 digits

Note: The synchronization interval between CA Harvest SCM and the CA SDM server is stored in the usdsynchinterval= parameter in the \CA_SCM_HOME\HBroker.arg file.

The LDAP Compliant Directory Configuration Windows

The LDAP Compliant Directory Configuration windows let you configure the LDAP settings for your CA Harvest SCM server.

Note: The product installation program records your responses to the prompts in the LDAP-related settings in the product configuration files HServer.arg and HBroker.arg.

This window contains the following fields:

LDAP Server Name

Defines one or more host names of the LDAP server to which your product computer connects, for example:

```
hostname1
```

You can optionally define the port number to use on each host, by entering the host name in the form *hostname*:*port*, for example:

```
hostname2:389
```

You can specify a list of host names separated by spaces. Each host may optionally be of the form *hostname*:*port*, for example:

```
hostname1:389 hostname2 hostname3:389
```

Important! If used, the :port number specified in the LDAP Server Name field overrides the value specified in the LDAP Port Number field.

Limits: 255 characters

If the host name field defines multiple host names, the product computer connects to the first available LDAP server in the list.

LDAP Port Number

Specifies the port number for the LDAP server. This port number is used if the LDAP port number is not specified in the host name field.

Default: If you are using SSL as the encryption mechanism, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

Base Distinguished Name

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
"ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Search Filter

(Optional) Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to the product, this filter is used to search for the user in the LDAP server.

Default: (&(objectclass=<*objectclass of user*>)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Example: Use -ldapattrusrname=uid as the LDAP server

In this example, the default search filter is used, and if -ldapattrusrname=uid is used for your LDAP server, then the search filter is the following:

```
(&(objectclass=person)(uid=<uid>))
```

Example: Use -ldapattrusrname=cn as the LDAP server

In this example, the default search filter is used, and if -ldapattrusrname=cn is used for your LDAP server, then the search filter is the following:

```
(&(objectclass=person)(cn=<cn>))
```

Example: Use -ldapattrusrname=uname as the LDAP server

In this example, the default search filter is used, and if -ldapattrusrname=uname is used for your LDAP server, then the search filter is the following:

```
(&(objectclass=person)(uname=<uname>))
```

Example: Use the Search Filter

In these examples, the default search filter, and the setting -ldapattrusrname=uid is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

- When the user amy33 attempts to log on, the search filter used to locate this user is the following:

  ```
  (&(objectclass=person)(uid=<amy33>))
  ```

- When the user john22 attempts to log on, the search filter used to locate this user is the following:

  ```
  (&(objectclass=person)(uid=<john22>))
  ```

LDAP Search Timeout

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Default: 60

Limits: 20 digits.

Username Attribute ID

Defines your LDAP server's LDAP user attribute name for a end-user user name.

Limits: 255 alphanumeric characters

LDAP/SASL Security/Encryption Mechanism

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security.

Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

None

Specifies no security mechanism.

Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

Default: None.

If you specify tls or ssl, complete the following fields; otherwise, skip them:

The TLS trusted certificate

(Optional) Defines the complete path name of the TLS trusted certificate file.

This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (the product remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

`TLS_CACERT` *`filename`*

Limits: 255 alphanumeric characters

The TLS client certificate

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CERT *filename*

Limits: 255 alphanumeric characters

The TLS client key

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the $HOME/.ldaprc file) using the following parameter:

TLS_KEY *filename*

Limits: 255 alphanumeric characters

Important! Private keys themselves are sensitive data and are typically password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

LDAP Distinguished Name

Defines the LDAP initial bind distinguished name (DN) to the LDAP server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
"cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Password for LDAP Distinguished Name

Defines the password for the LDAP distinguished name. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Your password is encrypted and is stored in the \CA_SCM_HOME\hserverauth.dfo file. This file name is specified in the following entry in the hserver.arg file:

```
ldapbindpwfile= hserverauth.dfo
```

Limits: 255 alphanumeric characters

Note: To update the encrypted password, use the command line utility svrenc. For more information about using svrenc, see the *Command Line Reference Guide*.

Fullname Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Full Name.

Limits: 255 alphanumeric characters

Phone Number Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Phone Number.

Limits: 255 alphanumeric characters

Phone Extension Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Phone Extension.

Limits: 255 alphanumeric characters

Fax Number Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Fax Number.

Limits: 255 alphanumeric characters

E-mail Address Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for eMail.

Limits: 255 alphanumeric characters

Synchronization frequency

Defines the authentication synchronization interval between the CA Harvest SCM broker and the authentication server. Use the input format *dd*[:*hh*[:*mm*[:*ss*]]], where *dd* is days, *hh* is hours, *mm* is minutes, and *ss* is seconds.

Default: 1 (1day)

Minimum: 0:1 (1hour)

Note: If the value of the authentication synchronization interval is invalid or less than one hour, the broker uses the minimum value (1 hour).

Limits: 20 characters

Examples:

-authsynchinterval=1:4 specifies 28 hours (1 day plus 4 hours).

-authsynchinterval=1:4:6 specifies 28 hours plus 6 minutes (1 day plus 4 hours plus 6 minutes).

-authsynchinterval=0:4:0:30 specifies 4 hours plus 30 seconds.

Note: When you install OpenLDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

You can optionally specify multiple base distinguished names when searching for user names in the LDAP server. To set up this capability, replace the existing description of the ldapbasedn=*base distinguished name* parameter with the following:

`ldapbasedn="name1[;name2[;name 3]…]"`

Defines one or more base distinguished names (DN) used when searching in the LDAP server.

To specify one base distinguished name, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com"
```

To specify two base distinguished names, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com;ou=europe,dc=abccorp,dc=com"
```

Important! When specifying multiple base distinguished names, separate them with a semicolon (;), as shown in the previous example.

Default: None

Limits: 255 characters

LDAP Parameters for External Usergroup Support

You can use the following parameters to define LDAP support for external user groups:

Note: Always enclose a value in quotation marks (" ") when it contains spaces.

-externalgroupenabled=1 or 0

(Optional) Use the following values to enable or disable your LDAP server's user group for external authentication:

- 1 enables your LDAP server's user group for external authentication.
- 0 disables your LDAP server's user group for external authentication.

-ldapgrpfilter=(&(objectclass=<objectclass of usergroup>)(usergroup-attribute-name=<placeholder>))

(Optional) Defines a group filter for locating a particular user group in the ldap server.

usergroup-attribute-name

Attribute of the usergroup used in -ldapattrusrgrpname=*attribute_name*.

<placeholder>

Identifies a literal constant placeholder for *usergroup-attribute-name*. Enter exactly the same value as *usergroup-attribute-name* and enclose the value with angle brackets (< >), as shown in the following example.

Example: Use -ldapattrusrgrpname=cn as the LDAP Server

In this example, if the objectclass of usergroup is group, and if -ldapattrusrgrpname=cn is used for your LDAP server, then the group filter is the following:

```
(&(objectclass=Group)(cn=<cn>))
```

-ldapattrusringrp=*attribute_name*

(Optional) Defines your LDAP server's attribute that evaluates members/users of a group, for example:

```
-ldapattrusringrp=member
```

-ldapattrgrpinusr=*attribute_name*

(Optional) Defines your LDAP server's attribute that evaluates groups of a user, for example:

```
-ldapattrgrpinusr=member_of
```

Set the License User Count

If your site has obtained an enterprise license certificate (key) for CA Harvest SCM, you must set the product broker to the correct maximum license user count before starting any product component. To work properly, the product requires that the maximum license user count matches the maximum user count for your license.

Important! If you have not received any licensing information, contact your account representative immediately to obtain a license.

Follow these steps:

1. Navigate to the %CA_SCM_HOME% directory and locate the hbroker.arg file.
2. Verify that the -maxserver parameter setting matches the maximum user count for your license.

   Note: The -maxserver parameter setting specifies the maximum number of product server processes permitted. One product server *process* corresponds to one product *user*. For example, if the maximum user count for your license is 100, set the -maxserver parameter to 100. The default user count is 50.
3. Save and close the hbroker.arg file.
4. Find and open the product's Lic98.log file, which typically exists in the C:\CA_LIC directory.
5. Check the file for an error message similar to the following message, which indicates that the -maxserver parameter is set higher than the licensed user count:

   ```
   CA Licensing -2CHA - Usage limitation exceeded. Contact your account representative to obtain a new license.
   ```
6. If this message does *not* appear, go to the next step.

   If this message appears, either lower the -maxserver count or contact your account representative to obtain a new license.
7. Copy the ca.olf file to the CA_LIC directory (typically C:\Program Files\CA\SharedComponents).

More information:

Broker Options (Windows) (see page 258)
Server Options (Windows) (see page 261)

Install the Broker as a Windows Service

By default, the CA Harvest SCM broker installs as a service, unless you clear the broker option during the custom install. If you did not clear the product broker as a Windows service during server installation, skip this procedure, because you do *not* need to install it again.

Note: For typical installations, the product broker is installed as a Windows service automatically.

Follow these steps:

1. If the database server resides on the same computer as the product server, enter the following at a command prompt:

   `bkrd.exe -install=`*`parent-service`*

   parent-service is the database service name for the DBMS you are using:

   - (Oracle) Specify OracleService*instance*, where *instance* is the name of your Oracle instance. For example, if the Oracle instance is named ORCL, enter the following command:

     ```
     bkrd.exe -install=OracleServiceORCL
     ```

   - (SQL Server 2005) Specify “SQL Server (instance)”, where instance is the name of your SQL Server *instance*. “SQL Server (MSSQLSERVER)” is typically the default. For example, enter the following command for the default instance:

     ```
     bkrd.exe -install="SQL Server (MSSQLSERVER)"
     ```

2. If the database server is on a different host than the product server, enter the following at a command prompt:

   ```
   bkrd.exe -install
   ```

3. Start the broker service in the Control Panel (Administrative Tools, Services). Start the service named CA Harvest SCM Broker Service.

4. To verify that the broker has started as a service, do the following:

   a. Start the Control Panel (Administrative Tools, Services).

   b. Check the Services list to confirm that the CA Harvest SCM Broker Service is listed. Confirm that the Status setting is Started and the Startup setting is Automatic.

   c. Close the Settings window and the Control Panel.

Note: After you initially install the broker as a service, the service does not start until you restart your system or start the service manually.

How to Verify and Configure the Installation

After you finish the CA Harvest SCM server installation, verify and configure your installation to help ensure that the installation was successful and that CA Harvest SCM runs efficiently.

Follow these steps:

1. Use the hdbsetup Database Configuration Utility to create or configure your product database.
2. Verify that the product server was installed successfully.
3. (Optional) Configure CA Software Delivery.
4. (Optional) Configure OpenLDAP authentication.

More information:

The hdbsetup Database Configuration Utility (see page 223)
How to Verify the Installation (see page 48)
How to Configure the CA Software Delivery Integration (Windows) (see page 49)
OpenLDAP Authentication Setup (see page 49)

How to Verify the Installation

Verifying the installation of the CA Harvest SCM server helps you find any potential problems before you start using the product.

Follow these steps:

1. If you performed a *typical* installation, restart your computer, open the Windows Services window, and verify that the product's Broker Service is started.
2. If you performed a *custom* installation and installed the product broker service, use one of the following methods to verify that the broker starts without errors:
   - Open the Windows Services window and verify that the product's Broker Service is running. If the service is not already running, start it.
   - Open the Windows Services window and verify that the product's Broker Service is *not* running. Next, open the Command Prompt window and enter **bkrd** at the command prompt.

   If you performed a custom installation and did *not* install the product's broker *service*, open the Command Prompt window and enter **bkrd** at the command prompt to start the broker.

3. After you start the broker, if a Licensing Manager Notification dialog appears displaying CA licensing details, do one of the following:
   - Wait for the dialog to close automatically.
   - Close the dialog manually and optionally specify that the dialog will not appear again.

How to Configure the CA Software Delivery Integration (Windows)

If you installed the CA Software Delivery Integration, before using it do the following:

- Verify that CA Software Delivery is installed and configured in your environment.

  Note: For more information, see your CA Software Delivery documentation.
- Review the information about how to use CA Harvest SCM and CA Software Delivery together to deploy software.

OpenLDAP Authentication Setup

Important! (Valid *only* if you installed OpenLDAP authentication.) Before you can use OpenLDAP authentication, you must configure the CA Harvest SCM components to use it.

The external authentication server should always have at least one user who is in the Administrator user group in CA Harvest SCM.

Note: The initial product user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of *1*. This user is always an administrator and always exists in the product, even if this user does not exist in the external authentication server. However, when you use external authentication, this user (like all other product users) must exist in the external authentication server to log in to the product.

More information:

External Authentication Configuration (see page 297)

How to Prepare for the Client Installation

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the client until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Become familiar with the different client installation options.
3. Determine the home directory (%CA_SCM_HOME%) in which you want to install the client. The default home directory is C:\Program Files\CA\SCM.
4. Decide whether you want to install the client on a network for shared use (see page 53), and determine where to install the shared client.

   Important! The shared location must be a network or local drive that is shared and to which other users can map. You cannot use a URL.

5. Decide whether you want to install the client in unattended (silent) mode (see page 55).
6. Determine whether you need to install the Administrator, in addition to installing the Workbench (see page 52).

Client Installation Options

You have the following options for installing the CA Harvest SCM client on Windows:

- Client with Workbench and Administrator.
- Client with Workbench Only.

The Client with Workbench Only option is a subset of the Client with Workbench and Administrator option. The Client with Workbench Only option contains all the client features except Administrator.

Usually, you should select the Workbench and Administrator option, which includes both Workbench and Administrator. However, if you do not want to make the Administrator available as an optional feature during the installation, select the Workbench Only option.

Important! Do *not* mix the two options on a single computer. For example, if you have installed the client using the Workbench and Administrator option, and you later want to switch and use the Workbench only option, first uninstall the existing client. Then, install the client again using the Workbench only option.

Install the Local Client

To successfully install or upgrade the CA Harvest SCM client, you must have Windows Administrator rights.

Follow these steps:

1. Insert the installation media into your drive.
2. The Product Explorer dialog appears.
3. Select the product component that you want to install.

   Note: If your computer has a previous release of the product installed, you are prompted to remove it. You must uninstall the previous release before you can install the new release.
4. To continue with the installation, follow the on-screen instructions.
5. When prompted, enter your user name and organization name, and specify who can use the product.

   Note: The Only for me (CA User) option applies for a client only installation.
6. When prompted, specify the location to install the product. By default, the product is installed to the C:\Program Files\CA directory.

   Note: If you change the default installation directory, the directory you specify will be the top-level directory for the product files. For example, %CA_SCM_HOME% will be defined as *your-installation-path*\SCM. If the directory you specify does not exist, it is created.

   Consider the following information when specifying the location to install the product:

   - When upgrading and when prompted for the installation path, do *not* select the %CA_SCM_HOME% location *path*\CCC_Harvest from a previous installation. Instead, either select the default location path (C:\Program Files\CA) or specify a new location.
   - If you specify a non-default installation path that includes an ampersand (&) in a folder name, the ampersand is included in the path name during installation. However, the ampersand does not appear in the path name in the destination folder during installation. Instead, the ampersand is removed and the next character after it is an underline. For example, if the folder name is CA&SCM, CA_SCM is displayed. However, the folder name used for the installation is CA&SCM.
   - If you have installed another product component on this computer, such as the server or agent, you must install the client in the same location as the existing component.
   - The plug-in for Eclipse is installed and maintained separately from the product components.

7. When prompted, select how you want to install the client by selecting one of the following options.

   Typical

   Installs the client with the most popular features.

   Custom

   Select optional features for your client installation (see page 52).

8. Continue following the on-screen instructions to complete the client installation.

   In addition to the features you have selected, the following CA shared components are automatically installed:

   - Enterprise Communicator (PEC). If you want to uninstall this component, you must use the Windows Add/Remove control panel. You cannot uninstall PEC if the product is still installed.
   - The Public Key Infrastructure (eTPKI). This component is installed automatically through the product installation.

   Note: Because these components are shared by other CA products, they are not removed when the product is uninstalled. They are separate components in the Windows Add/Remove control panel and must be removed separately.

Note: If you specify a nondefault installation path that includes an ampersand (&) in a folder name, the ampersand is included in the path name during installation. However, the ampersand does not appear in the path name in the destination folder during installation. Instead, the ampersand is removed and the next character after it is an underline. For example, if the folder name is CA&SCM, CA_SCM is displayed. However, the folder name used for the installation is CA&SCM.

Client Installation (Optional Features)

When you install the client on Windows, you can selectively select to install the following optional features:

Workbench

Installs the GUI application that provides access to all product user functions.

Administrator

Installs the Administrator. This option is nested under the Workbench, because the Administrator requires the Workbench to work properly. By default, the Administrator is installed with the Workbench. If you do not want to install the Administrator, expand the Workbench option and clear the Administrator.

Important! If you need to install the Administrator, in addition to the Workbench, run setup.exe from the \bin\win32_client directory on the installation media. If not, run setup.exe from the \bin\win32_workbench directory on the installation media.

Documentation

Installs the complete product documentation set.

Command-Line Utilities

Installs a group of command-line applications that you can use to perform operations from the command line.

Windows Shell Extension

Enables access to the product's version control system using the Windows Explorer menus.

How to Install the Client on a Network

Use the Network installation option to run the client from a network location. You can select features to run from the network, or features to install to the local computer. If you decide to run the client from the network, all features you select must be run from the network. If you decide to install the client locally, then all the features you install locally must be run locally. This setup uses minimal file space on the local client computer and executes the product from a shared location on the network.

Follow these steps:

1. Set up the shared client on the network location.
2. Install the client to local computers from the network location.

Note: You must have Windows Administrator rights to use the Network installation option.

Set Up the Shared Client

This step should be completed by your CA Harvest SCM Administrator.

Important! The shared location must be a network or local drive that is shared and to which other users can map. You cannot use a URL.

Follow these steps:

1. Open a command prompt and navigate to the location of the product installation files. For example:

   `cd DVD-drive:\bin\win32_client`

2. At the command prompt, enter the following command:

   `setup.exe /a`

   In this command, the /a option specifies an administrative installation. The installation wizard starts and prompts you for an installation path.

3. Manually enter the network location to which a server image of the product will be created, or click Change to navigate to a different location.

   Note: This location must be a network or a local drive that is shared and to which other users can map. You *cannot* use a URL. If you already have a shared network agent, you must use the same network location for the client.

4. Continue following the on-screen instructions to set up the shared client.

5. Verify that the shared directory and sub-directories permit read-only access for users to perform the network client installation.

6. Communicate the location of the network installation to all users who will be installing the client from this network location.

Install the Client from the Network

Complete this step on each end-user computer. The network client installation requires that the shared directory be created.

Follow these steps:

1. On the network server, map a drive to the shared CA Harvest SCM directory location.

2. Typically, enter the following on the network computer and share it:

   `C:\apps\ca\SCM`

3. On the end-user computer, map to the share. For example, if the share is \\hostname\apps, map the local share *N:* to \\hostname\apps.

4. On the end-user computer, use Windows Explorer to browse to the share and double-click the file named CA Harvest Software Change Manager Client.msi.

   Important! The drive mapping applies only to the current user who is installing the client. If, at a later time, a different user logs in to the same computer, they must map to the shared directory to use the product.

5. Install the client using the Custom installation option and select the features you want to run from the network. Do *not* mix network client features with local client features.

   Note: The icon changes to a network icon when you select network client features.

6. When prompted, specify whether to install Enterprise Communicator (PEC) on the network or on the local computer.

   If you specify that PEC is to be installed on the network, only the environment variables and registry entries necessary for PEC to function are copied and set up on the local computer. This option is the default.

   If you specify that PEC is to be installed on the local computer, all PEC files are installed locally.

7. When the installation is finished, optionally restart your computer if prompted.

   Note: If you select all features to run from the network, the Destination Folder will be created but will be empty. Do not delete this folder.

Verify the Client Installation

Verify that the client has been installed properly.

To verify that the client was installed successfully, start the client.

The client appears, which verifies that it is installed properly.

How to Install the Client Silently

You can perform a manual unattended (silent) client installation using the command line. You can use the command line installation for first-time installations only, not upgrades.

Use the following syntax to perform an unattended installation of the client from the command line:

```
DVD-drive:\install-path\setup.exe /s /v"/qoption [property1=\"value\"
property2=\"value\" property3=\"value\"...]"
```

install-path

Specifies one of the following directories on the installation media:

- To install the client with both Workbench and Administrator:

  `\bin\win32_client`

- To install the client with Workbench only:

  `\bi4n\win32_workbench`

/s

Specifies a silent installation, requiring no response after the installation is started. If you do not specify the /s argument, dialogs will appear during the installation, requiring user response.

/v

Passes command line switches and values of public properties to msiexec.exe. Any quotation marks inside the value for the /v parameter must be preceded by a backward slash \ (\").

/qoption

Specifies the options for setting what kind of user interface (UI) appears during the installation, as follows:

q, qn

No UI.

qb

Basic UI. Use qb! to hide the Cancel button.

qr

Reduced UI with no modal dialog displayed at the end of the installation.

qf

Full UI with any authored FatalError, UserExit, or Exit modal dialogs at the end.

qn+

No UI except for a modal dialog displayed at the end.

qb+

Basic UI with a modal dialog displayed at the end. The modal dialog appears if the user cancels the installation. Use qb+! to hide the Cancel button.

qb-

Basic UI with no modal dialogs; qb+- is not a supported UI level. Use qb-! to hide the Cancel button.

qb+!, qb-!

Basic UI with or without the modal dialog displayed at the end. Hides the Cancel button. These options can also be entered qb!+ and qb!-.

Note: The ! option is available with Windows Installer version 2.0 and works only with basic UI. It is not valid with the full UI.

property1=\"value\" property2=\"value\" property3=\"value\"...

Specifies one or more of the installation public properties. If there are spaces within the value, enclose the value in quotation marks, which must be preceded by the backslash character, as shown in this statement. For example, INSTALLDIR=\"C:\program files\CA\" specifies a path name with spaces, and COMPANYNAME=\"Jones Consulting Firm\" specifies a company name with spaces.

To clear a public property using the command line, set its value to an empty string.

Statements include the following:

INSTALLDIR=\"*directory*\"

Specifies the target installation directory for the client.

USERNAME=\"*name*\"

Specifies the name of the user who will be using the client.

COMPANYNAME=\"*name*\"

Specifies the name of the company for which the user works.

ALLUSERS=\"*value*\"

If this installation is for the current user *only*, do *not* set this property from the command line or set its value to an empty string.

1 makes the client available to all users who use this computer.

2 makes the client available to the current user only if the current user does not have admin rights; otherwise, install to all users.

ADDLOCAL=\"*value*\"

Specifies a list of features delimited by commas that are to be installed locally. The following lists the available features. Items marked with an asterisk (*) are always installed if ADDLOCAL or ADDSOURCE is *not* specified. However, if ADDLOCAL or ADDSOURCE is specified, you specify individually each feature that you want to install from the following lists. The feature names are case-sensitive. Use only a comma (no space) to separate the features.

- **For client installation**-Administrator*, Workbench*, Documentation*, CommandLine*, Windows Shell Extensions (HarWin).
- **For Workbench only installation**-Workbench*, Documentation*, CommandLine*, Windows Shell Extensions (HarWin).

Note: Use the feature name as Client64 instead of Workbench, for 64 bit client installation and workbench installations *only*.

ADDSOURCE=\"*value*\"

Specifies a list of features delimited by commas that are to be installed to run from the source. Typically, you use the ADDSOURCE parameter to install the client on a network. The same values and rules for specifying values that apply to the previous parameter (ADDLOCAL) also apply to ADDSOURCE.

INSTALLPECTONETWORK=\"Yes|No\"

Specify Yes to install PEC on the network or specify No to install PEC on the local computer. The default value is Yes.

This variable takes effect for network installations *only*; that is, if ADDSOURCE is specified.

/l \"path-name\log-file-name.log\"

The lowercase L operand creates the log file named log-file-name.log in the location you specify (path-name). The log file records all installation actions and any errors encountered during the installation process. For example, the following command specifies that the log file client.log be created in the C:\scm Downloads directory:

```
setup.exe /s /v"/l \"C:\scm Downloads\client.log\"
```

Important! If the computer needs to be restarted after a silent installation, Windows Installer restarts the computer with no warning.

How to Prepare for the Agent Installation

Follow these steps:

1. If you have not already done so, read the *Release Notes*. Do not install the agent until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Determine the home directory (%CA_SCM_HOME%) in which you want to install the agent. The default home directory is C:\Program Files\CA\SCM.

3. Determine the agent port number.

4. Decide the method (either internal or OpenLDAP) the product should use to authenticate users' logon credentials.

5. Decide whether you want to install the agent on a network for shared use (see page 67), and determine where to install the shared agent.

   Important! The shared location must be a network or local drive that is shared and to which other users can map. You cannot use a URL.

6. Decide whether you want to install the agent in unattended (silent) mode. (see page 69)

7. Determine whether you installed the agent as a Windows service (see page 77).

Authentication

During the CA Harvest SCM installation, you can select to use either *internal* authentication, *OpenLDAP* authentication, or *Mixed Mode Authentication* as the method your site will use to authenticate users' names and passwords. The authentication is used, for example, when a user attempts to log in to the product.

- Internal authentication uses the product to authenticate the user name and password.
- OpenLDAP authentication uses an OpenLDAP authentication server for authentication.

- Mixed Mode authentication lets the SCMAdmin create users internally even though the authentication mode may be set to External (LDAP).

  Note: Mixed Mode authentication does not use LDAPserver for Authentication when users are created internally.

The authentication method that you select may depend on your company's IT standards and conventions, resources, environment-specific concerns, manager input, in addition to other factors.

If you select internal authentication, you do not need to perform any preparation tasks.

How to Prepare for the OpenLDAP Installation

Based on your planned security-related implementation, you may decide to use OpenLDAP authentication instead of internal authentication.

Follow these steps:

1. Verify that your LDAP server is installed and configured.

   Note: For a list of supported LDAP servers, read the Release Notes.

2. Decide whether to use Transport Level Security (TLS), Secure Socket Layer (SSL), or no security to encrypt communication between the product and your LDAP server. Specify TLS *only* if your LDAP server supports StartTLS.

   Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

3. If you select TLS or SSL, determine and record complete path names for the following:
   - The TLS trusted certificate (optional)
   - The TLS client certificate (optional)
   - The TLS client key (optional)
4. If you select TLS or SSL, decide which method you will use to supply the TLS values when you are asked to supply them during the installation:
   - By entering the actual path names.
   - By entering the name of the OpenLDAP configuration file that specifies these path names.

More information:

External Authentication Configuration (see page 297)

Install the Agent

The CA Harvest SCM agent acts as a file server to remote computers. You can install *only* the agent, without installing the product server or client.

Follow these steps:

1. Insert the installation media into your drive.

   The Product Explorer dialog appears.

2. Select the product component that you want to install.

3. To continue with the installation, follow the on-screen instructions.

4. When prompted, enter your user name and organization name, and specify who can use the product.

5. When prompted, specify the location to install the product. By default, the product is installed to the C:\Program Files\CA directory.

   Note: If you change the default installation directory, the directory you specify will be the top-level directory for the product files. For example, %CA_SCM_HOME% will be defined as *your-installation-path*\SCM. If the directory you specify does not exist, it is created.

   Consider the following information when specifying the location to install the product:

   - When upgrading and when prompted for the installation path, do *not* select the %CA_SCM_HOME% location *path*\CCC_Harvest from a previous installation. Instead, either select the default location path (C:\Program Files\CA) or specify a new location. The installation path name is the location for CA products, *not* %CA_SCM_HOME%. The %CA_SCM_HOME% is located one directory level under the installation path.

   - If you specify a non-default installation path that includes an ampersand (&) in a folder name, the ampersand is included in the path name during installation. However, the ampersand does not appear in the path name in the destination folder during installation. Instead, the ampersand is removed and the next character after it is an underline. For example, if the folder name is CA&SCM, CA_SCM is displayed. However, the folder name used for the installation is CA&SCM.

6. When prompted, select how you want to install the agent by selecting one of the following options.

   Complete

   Installs the agent with the most popular features.

Custom

Select optional features for your agent installation.

7. When prompted, specify the agent port number.
8. When prompted specify if the Agent should be started as a service (Custom Install only).
9. When prompted, specify whether to enable or disable FIPS mode.
10. When prompted, specify the method the product will use to authenticate users' logon credentials:

    Internal

    This method uses operating system calls. Login credentials provided to the remote agent are validated against the remote agent's operating system. If you select Internal, skip the rest of this step and continue at the next step.

    OpenLDAP

    This method uses an external server. Login credentials provided to the remote agent are validated against the external authentication server. If you select OpenLDAP authentication, you are prompted to supply the required LDAP-related information (see page 62). When you are finished specifying LDAP information, continue at the next step.

    Note: When you install LDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

11. Continue following the on-screen instructions to complete the agent installation.

    In addition to the features you have selected, the following CA shared components are automatically installed:

    - Enterprise Communicator (PEC). If you want to uninstall this component, you must use the Windows Add/Remove control panel. You cannot uninstall PEC if the product is still installed.
    - The Public Key Infrastructure (eTPKI). This component is installed automatically through the product installation.

    Note: Because these components are shared by other CA products, they are not removed when the product is uninstalled. They are separate components and must be removed separately.

LDAP Compliant Directory Configuration Windows

The LDAP Compliant Directory Configuration windows let you configure the LDAP settings for your CA Harvest SCM agent. The window uses the following fields.

Note: The product installation program records your responses to the following prompts in the LDAP-related settings in the product configuration files HServer.arg, HBroker.arg, and HAgent.arg.

LDAP Server Name

Defines one or more host names of the LDAP server to which your CA Harvest SCM computer connects, for example:

```
hostname1
```

You can optionally define the port number to use on each host, by entering the host name in the form *hostname*:*port*, for example:

```
hostname2:389
```

You can specify a list of host names separated by spaces. Each host may optionally be of the form *hostname*:*port*, for example:

```
hostname1:389 hostname2 hostname3:389
```

Important! If used, the :port number specified in the LDAP Server Name field overrides the value specified in the LDAP Port Number field.

Limits: 255 characters

If the host name field defines multiple host names, the product computer connects to the first available LDAP server in the list.

LDAP Port Number

Specifies the port number for the LDAP server. This port number is used if the LDAP port number is not specified in the host name field.

Default: If you are using SSL as the encryption mechanism, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

Base Distinguished Name

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
"ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Search Filter

(Optional) Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to the product, this filter is used to search for the user in the LDAP server.

Default: (&(objectclass=person)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Examples

These examples use the default search filter.

If -ldapattrusrname=uid for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uid=<uid>))
```

If -ldapattrusrname=cn for your LDAP server, then the search filter is:

```
(&(objectclass=person)(cn=<cn>))
```

If -ldapattrusrname=uname for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uname=<uname>))
```

Examples: How the Search Filter is Used

The search filter is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

These examples use the default search filter and use the setting -ldapattrusrname=uid:

When the user amy33 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<amy33>))
```

When the user john22 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<john22>))
```

LDAP Search Timeout

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Default: 60

Limits: 20 digits.

Username Attribute ID

Defines your LDAP server's LDAP user attribute name for a user's user name.

Limits: 255 alphanumeric characters

LDAP/SASL Security/Encryption Mechanism

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security.

Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

None

Specifies no security mechanism.

Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

Default: None.

If you specify tls or ssl, complete the following fields; otherwise, skip them:

Trusted Certificate Filename

(Optional) Defines the complete path name of the TLS trusted certificate file.

This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (the product remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CACERT *filename*

Limits: 255 alphanumeric characters

Client Certificate Filename

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CERT *filename*

Limits: 255 alphanumeric characters

Client Key Filename

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the $HOME/.ldaprc file) using the following parameter:

TLS_KEY *filename*

Limits: 255 alphanumeric characters

Important! Private keys themselves are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

LDAP Distinguished Name

Defines the LDAP initial bind distinguished name (DN) to the LDAP Server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
"cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Password for LDAP Distinguished Name

Defines the password for the LDAP distinguished name. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Limits: 255 alphanumeric characters

Your password is encrypted and is stored in the \CA_SCM_HOME\hagentauth.dfo file. This file name is specified in the following entry in the hagent.arg file:

```
ldapbindpwfile= hagentauth.dfo
```

You can optionally specify multiple base distinguished names when searching for user names in the LDAP server. To set up this capability, replace the existing description of the ldapbasedn=*base distinguished name* parameter with the following:

ldapbasedn="*name1[;name2[;name 3]…]*"

Defines one or more base distinguished names (DN) used when searching in the LDAP server.

To specify one base distinguished name, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com"
```

To specify two base distinguished names, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com;ou=europe,dc=abccorp,dc=com"
```

Important! When specifying multiple base distinguished names, separate them with a semicolon (;), as shown in the previous example.

Default: None

Limits: 255 characters

OpenLDAP Authentication (Agent Installation) Configuration

Important! (Valid *only* if you installed OpenLDAP authentication.) Before you can use OpenLDAP authentication, you must configure the CA Harvest SCM components to use it.

The external authentication server should always have at least one user who is in the Administrator user group in CA Harvest SCM.

Note: The initial product user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of *1*. This user is always an administrator and always exists in the product, even if this user does not exist in the external authentication server. However, when you use external authentication, this user (like all other product users) must exist in the external authentication server to log in to the product.

How to Install the Agent on a Network

Use the Network installation option to run the agent from a network location. This setup uses minimal file space on the local client computer and executes the product from a shared location on the network.

Follow these steps:

1. Set up the shared agent in the network location.
2. Install the agent to local computers from this network location.

Note: You must have Windows Administrator rights to use the Network installation option. In addition, if you installed the agent on a network, you cannot run the agent as a Windows service.

Set Up the Shared Agent

This step should be completed by your CA Harvest SCM Administrator.

Follow these steps:

1. Open a command prompt and navigate to the location of the agent installation files. For example:

   ```
   cd DVD-drive:\bin\win32_agent
   ```

2. At the command prompt, enter the following command:

   ```
   setup.exe /a
   ```

   In this command, the /a option specifies an administrative installation. The installation wizard starts and prompts you for an installation path.

3. Manually enter the network location to which a server image of the product will be created, or click Change to navigate to a different location.

   Note: This location must be a network or a local drive that is shared and to which other users can map. You *cannot* use a URL. If you already have a shared network client, you must use the same network location for the agent.

4. Continue following the on-screen instructions to set up the shared agent.

5. Verify that the shared directory and sub-directories permit read-only access for the network users to perform the network agent installation.

6. Communicate the location of the network installation to all users who will be installing the agent from this network location.

Install the Agent from the Network

Complete this step on each end-user computer. The network agent installation requires that the shared directory be created.

Follow these steps:

1. On the end-user computer, map to the shared network location. For example, if the share is \\hostname\apps, map the local share N: to \\hostname\apps.
2. On the end-user computer, use Windows Explorer to browse to the share and double-click the file named CA Harvest Software Change Manager Agent.msi.

   Important! The drive mapping applies only to the current user who is installing the product. If, at a later time, a different user logs into the same computer, they must map to the shared directory to use the product.
3. Install the agent using the Custom installation option and select the agent to run from the network.
4. When the installation is finished, optionally restart the computer if prompted.

   Note: If you select all features to run from the network, the Destination Folder will be created but will be empty. Do *not* delete this folder.

How to Install the Agent Silently

You can perform an agent installation using the command line instead of the installation wizard. You can use the command line for first-time installations *only*, not upgrades. Use the following syntax to perform an unattended installation of the agent from the command line:

DVD-drive:\bin\win32_agent\setup.exe /s /v"/q*option* [*property1*=\"*value*\" *property2*=\"*value*\" *property3*=\"*value*\"...]"

/s

Specifies a silent installation, requiring no response after the installation is started. If you do not specify the /s parameter, dialogs appear during the installation, requiring user response.

/v

Passes command line switches and values of public properties through to msiexec.exe. Any quotation marks inside the value for the /v parameter must be preceded by a backward slash \ (\").

/qoption

Specifies the options for setting what kind of user interface (UI) appears during installation, as follows:

q, qn

No UI.

Qb

Basic UI. Use qb! to hide the Cancel button.

Qr

Reduced UI with no modal dialog displayed at the end of the installation.

Qf

Full UI with any authored FatalError, UserExit, or Exit modal dialogs at the end.

qn+

No UI except for a modal dialog displayed at the end.

qb+

Basic UI with a modal dialog displayed at the end. The modal dialog does appear if the user cancels the installation. Use qb+! to hide the Cancel button.

qb-

Basic UI with no modal dialogs; qb+- is not a supported UI level. Use qb-! to hide the Cancel button.

qb+!, qb-!

Basic UI with or without the modal dialog displayed at the end. Hides the Cancel button. These options can also be entered qb!+ and qb!-.

Note: The ! option is available with Windows Installer version 2.0 and works only with basic UI. It is not valid with the full UI.

property1=\"value\" property2=\"value\" property3=\"value\"...

Specifies one or more of the installation public properties. If there are spaces within the value, enclose the value in quotation marks, which must be preceded by the backslash character, as shown in this statement. For example, INSTALLDIR=\"C:\program files\CA\" specifies a path name with spaces, and COMPANYNAME=\"Jones Consulting Firm\" specifies a company name with spaces.

To clear a public property using the command line, set its value to an empty string.

INSTALLDIR=\"*directory*\"

Specifies the target installation directory for the agent.

SCMAGENTPORT=\"*port_number*\"

Specifies the agent port number.

CREATEAGENTSERVICE=\"*value*\"

The default is to create a Windows service. Specify No to not create a Windows service for this agent.

USERNAME=\"*name*\"

Specifies the name of the user who will be using the agent.

COMPANYNAME=\"*name*\"

Specifies the name of the company for which the user works.

ALLUSERS=*value*

If this installation is for the current user *only*, do *not* set this property from the command line or set its value to an empty string.

1 makes the agent available to all users who use this computer.

2 makes the agent available to the current user only if the current user does not have admin rights; otherwise, install to all users.

ADDLOCAL=Agent

Specifies that the agent should be installed locally.

ADDSOURCE=Agent

Specifies that the agent should be run from the source location, typically used for a network installation.

Authentication Options

Use the following property=\"value\" statements to specify the type of authentication to use for validating users' logon credentials, as follows:

- To use internal authentication, specify only the AUTHMODE=internal parameter and do not specify the remaining LDAP authentication parameters in this section.
- To use LDAP authentication, specify the following parameters. Enter the back slashes (\) and quotation marks (" ") literally as shown.
  - AUTHMODE=\"openldap\"
  - UPDATELDAPSETTING=\"Yes\"
  - All applicable LDAP authentication parameters in this section

Important! To use LDAP authentication, you must specify both the AUTHMODE=\"openldap\" and the UPDATELDAPSETTING=\"Yes\" properties; otherwise, the remote agent uses the default value, internal authentication, instead of LDAP authentication.

Note: For more information about LDAP authentication, see Install the Agent.

AUTHMODE=\"internal|openldap\"

Specifies what type of authentication to use:

internal

Uses operating system calls. Login credentials provided to the remote agent are validated against the remote agent's operating system If you specify internal, skip the remaining LDAP option entries.

openldap

Specifies OpenLDAP authentication. Login credentials provided to the remote agent are validated against the external authentication server.

Important! If you specify openldap, specify the remaining LDAP entries.

UPDATELDAPSETTING=\"Yes\"

To use OpenLDAP authentication, you must specify *both* the AUTHMODE=\"openldap\" and UPDATELDAPSETTING=\"Yes\" settings.

LDAPSERVER=\"hostname[:portnumber]\"

Defines one or more host names of the LDAP server to which this agent connects, for example:

LDAPSERVER=\"hostname1\"

You can optionally define the port number to use on each host, by entering the host name in the form *hostname*:*port*, for example:

LDAPSERVER=\"hostname2:389\"

Note: Enter the back slashes (\) and quotation marks (" ") literally as shown.

You can specify a list of host names-separated by spaces and enclosed in quotation marks. Each host may optionally be of the form *hostname*:*port*, for example:

LDAPSERVER=\"hostname1:389 hostname2 hostname3:389”\"

Important! If used, the :port option overrides the port number provided in the LDAPPORT=\"portnumber\" parameter.

Limits: 255 characters

If the host name field defines multiple host names, the server or agent connects to the first available LDAP server in the list.

LDAPPORT=\"portnumber\"

Specifies the port number for the LDAP server computer. This port number is used if the LDAP port number is not specified in the LDAPSERVER=\"*hostname*[:*portnumber*]\" parameter.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Default: If the security mechanism is ssl, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

LDAPSEARCHTIMEOUT=\"seconds\"

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Default: 60

Limits: 20 digits.

LDAPBINDDN=\"distinguished-name\"

Defines the LDAP initial bind distinguished name (DN) to the LDAP server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
"cn=john22,ou=users,ou=north america,dc=abccorp,c=com"
```

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

LDAPBINDPW=\"password\"

Defines the password for the LDAP distinguished name. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Your password is encrypted and is stored in the \CA_SCM_HOME\hagentauth.dfo file. This file name is specified in the following entry in the hagent.arg file:

```
ldapbindpwfile= hagentauth.dfo
```

Limits: 255 alphanumeric characters

LDAPBASEDN=\"base-distinguished-name\"

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
"ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

LDAPMODE=\"{none|tls|ssl}\"

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security. Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

None

Specifies no security mechanism.

Important! If you specify **tls** or **ssl**, specify the following parameters: TLSTRCERTFILE=, TLSCERTFILE=, and TLSKEYFILE=. For complete details about these TLS values, including how to specify them during the installation using either a file name or an LDAP configuration file, see the descriptions of the TLS fields in LDAP Compliant Directory Configuration Window.

TLSTRCERTFILE=\"*filename*\"

(Optional) Defines the complete path name of the TLS trusted certificate file.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Limits: 255 alphanumeric characters

TLSCERTFILE=\"*filename*\"

(Optional) Defines the complete path name of the TLS client certificate file.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Limits: 255 alphanumeric characters

TLSKEYFILE=\"*filename*\"

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Limits: 255 alphanumeric characters

Important! Private keys themselves are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

FIPSMODE=*value*

Enables or disables FIPS 140-2, which is an encryption standard that protects data from unauthorized programs and users. This option enforces compliance to FIPS 140-2, which has a specific set of standards for cryptographic modules. Use the following values to disable or enable the FIPS mode:

- **0** disables the FIPS mode.
- **1** enables the FIPS mode.

Default: 0

Limits: 1 character

LDAP User Search Filter

You can use an RFC-2254-compliant search filter for locating a user.

LDAPFILTER=\"*search-filter*\"

(Optional) Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to the product, this filter is used to search for the user in the LDAP server.

Enter the back slashes (\) and quotation marks (" ") literally as shown.

Default: (&(objectclass=person)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Examples

These examples use the default search filter.

If -ldapattrusrname=uid for your LDAP server, then the search filter is:

(&(objectclass=person)(uid=<uid>))

If -ldapattrusrname=cn for your LDAP server, then the search filter is:

(&(objectclass=person)(cn=<cn>))

If -ldapattrusrname=uname for your LDAP server, then the search filter is:

(&(objectclass=person)(uname=<uname>))

Examples: How the Search Filter is Used

The search filter is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

These examples use the default search filter and use the setting -ldapattrusrname=uid:

When the user amy33 attempts to log on, the search filter used to locate this user is:

(&(objectclass=person)(uid=<amy33>))

When the user john22 attempts to log on, the search filter used to locate this user is:

(&(objectclass=person)(uid=<john22>))

Install the Agent as a Windows Service

Skip this procedure if either of the following applies:

- You *already* installed the CA Harvest SCM agent as a Windows service during the agent installation. In this case, you do *not* need to perform these steps.
- You installed the agent on the *network.* You *cannot* run the agent as a Windows service when the agent is installed on the network.

If neither applies, consider the following information and follow these steps to install the agent as a service.

- You must have Windows Administrator rights to install the agent.
- After you install the agent as a service, the service does not *start* until you restart your computer or manually start the service.

Follow these steps:

1. Enter the following from a command prompt:

   ```
   agntd.exe -install
   ```

2. Start the Control Panel (Administrative Tools, Services).
3. Start the service named CA Harvest SCM Agent Service.

How to Verify the Agent Service Status

Follow these steps:

1. Start the Control Panel (Administrative Tools, Services).
2. Check the Services list to confirm that the CA Harvest SCM Agent Service is listed. Confirm that the Status setting is Started and the Startup setting is Automatic.
3. Close the Settings window and the Control Panel.

Chapter 3: Installing on UNIX, Linux, and zLinux

Important! Installing these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

How to Prepare for the Server Installation

Important! On UNIX and Linux, if you install the product server, the product client (command-line utilities) and product agent are also installed automatically, whether the database is local or remote.

Note: CAcrypto installation is not required for a new Release 12.5 install. CAcrypto installation is used only when upgrading from an existing Release 7.1 installation and to convert the Release 7.1 .dfo file to Release 12.5 format.

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the server until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Verify that you own the files and directories.

3. Decide whether to perform a *typical* installation or a *custom* installation.
   - Select the *typical* option to do the following:
     - Install the product using predefined settings without providing any additional input.
     - Install the server, the command-line utilities, and product documentation.
     - Install these common, shared CA components: Enterprise Communicator (PEC), CA ODBC, eTPKI, and CA Licensing.
     - Automatically use internal authentication for product users.
   - Select the *custom* option to:
     - Configure settings in each step of the installation process.
     - Use Oracle for your product database.
     - Decide which components to install, including the server, command-line utilities, and product documentation.
     - Install these common, shared CA components: Enterprise Communicator (PEC), CA ODBC, eTPKI, and CA Licensing
     - Install CA Software Delivery.
     - Enable Federal Information Processing Standard (FIPS) mode for the product agent.
     - Specify a firewall port range.
     - Decide whether to use internal or OpenLDAP authentication for product users.
4. Verify that you know the installation directory paths for your DBMS, CAI/PT ODBC, and Enterprise Communicator (PEC) if they are not already set in your environment.
5. Verify that you know the CA Harvest SCM database user. You must specify a user to have the required access rights to the product tables. If your database is Oracle, then the product table owner is the product database user. You can assign this user any valid user name for the version of Oracle you are using.
6. Verify that you know the DBMS system account, because you are prompted for the password to the Oracle system account. The DBMS system account is used to log into the DBMS during installation.
7. If you plan to use *Oracle* as your product database, consider the following information:
   - You must install Oracle before using any product component, including the Database Configuration Utility. You must run this utility to set up your product database on Oracle before you can use the product to create users or to check in and check out files. We recommend that you install Oracle before installing the product, so that you can configure your product database immediately after installing the product server.

- If you use Oracle and Tomcat, do not use the same port number for the Tomcat and Oracle database servers. By default, the Tomcat server uses port 8080, and if the Oracle database also uses the same port, then an error will result.
- Do *not* use configdsn to set up your CA Harvest SCM database. You must use the Database Configuration Utility instead.

8. If you plan to use a *remote* Oracle database, consider the following information:
   - Verify that the Oracle client networking utilities are installed.
   - Verify that you understand how a product server using a remote Oracle database connects to a product database using Oracle client TCP/IP utilities.
   - Verify that a version of Oracle supported by the product is installed on the remote computer.

   Note: For information, see your Oracle documentation and the Release Notes.
9. Install CA Licensing (Lic98) software on the product server computer.
10. Install the Public Key Infrastructure (eTPKI) to help ensure the security of users, data, and applications in your enterprise.
11. Create the SCM user and the default installation directories.
    Install CAI/PT ODBC, the Open Database Connectivity (ODBC) driver. CAI/PT ODBC is required for product database communication.
12. Install Enterprise Communicator (PEC), a communications toolkit required for the product client/server communication.
13. Determine whether to install CA Software Delivery.
14. Select an authentication method, either internal authentication (CA Harvest SCM) or OpenLDAP.
15. Extract the installation files.

More information:

Install Lic98 Licensing (see page 82)
Create the Product User and Default Directories (see page 83)
Install CAI/PT ODBC (see page 89)
Install Enterprise Communicator (PEC) (see page 91)
Extract the Installation Files (see page 93)
The Public Key Infrastructure (see page 84)
Upgrade a Server (Local and Remote Oracle Database) (see page 203)

Install Lic98 Licensing

You can install Lic98 licensing on UNIX and Linux platforms. If you are *upgrading* to the current release from a previous release such as Release 12.x, you must install Lic98 licensing on the CA Harvest SCM server, as explained in the following procedure.

Important! Before starting these steps, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log on as the root user (the user named *root*).

Follow these steps:

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Copy the lic98.tar.gz file to a temporary location. For example, to copy the file to the /tmp directory, enter the following command:

   ```
   cp /cdrom/bin/directory/lic98.tar.gz /tmp
   ```

   directory

   Specifies the directory for your UNIX or Linux platform.
3. Change to the directory containing the lic98.tar file and extract the contents. For example, enter the following commands:

   ```
   cd /tmp
   gunzip lic98.tar.gz
   tar xvf lic98.tar
   ```
4. Enter the following command to change to the Lic98 product directory:

   ```
   cd lic98_install
   ```
5. The installation creates $CASHCOMP/ca_lic and $CASHCOMP/lib directories that contain the Lic98 files. The default location for the $CASHCOMP directory is /opt/CA/SharedComponents.

   If you are installing on a computer that has never had Lic98 installed and you want to change this location, change the $CASHCOMP environment variable from the default setting (/opt/CA/SharedComponents) to a new path name *before* running the installation. If the $CASHCOMP environment variable is not set, you can set it, according to the instructions provided by your UNIX or Linux operating system.

   Important! After Lic98 is installed on a computer, the Lic98 installation directory cannot be changed, even by updating the $CASHCOMP variable described in this step.
6. Enter the following command to run the installation:

   ```
   ./install
   ```

   The installation does not prompt you for any information.

Create the Product User and Default Directories

Important! Before creating the CA Harvest SCM user and default directories, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log on as the *root* user (the user named *root).*

Follow these steps:

1. Enter the following command to create the SCM group:

   ```
   groupadd cascm
   ```

2. Enter the following command to create a UNIX or Linux user named *cascm* who owns and runs the product server.

   ```
   useradd cascm
   ```

3. Enter the following command to add this user to the *cascm* group:

   ```
   usermod cascm -G cascm
   ```

4. Enter the following command to assign a password to this user.

   ```
   passwd cascm
   ```

   When prompted, specify the password.

5. If the product default directories do not already exist, enter the following commands to create them:

   ```
   mkdir /opt/CA/scm
   mkdir /opt/CA/caiptodbc
   mkdir /opt/CA/pec
   mkdir /opt/CA/etpki
   ```

   The default install location for the ODBC driver is in the /opt/CA/caiptodbc directory.

6. Enter the following commands to help ensure that the product user has write access to the required directories:

   ```
   chmod 775 /opt/CA/scm /opt/CA/etpki /opt/CA/caiptodbc /opt/CA/pec
   ```

7. Enter the following commands to help ensure that the SCM group owns the following directories:

   ```
   chgrp -R cascm /opt/CA/scm
   chgrp -R cascm /opt/CA/etpki
   chgrp -R cascm /opt/CA/caiptodbc
   chgrp -R cascm /opt/CA/pec
   ```

8. Enter the following commands to help ensure that the SCM user owns the following directories:

```
chown -R cascm /opt/CA/scm
chown -R cascm /opt/CA/etpki
chown -R cascm /opt/CA/caiptodbc
chown -R cascm /opt/CA/pec
```

9. As the new SCM UNIX or Linux user (su cascm), complete these tasks:

   - If you are using Oracle, create the environmental variables ORACLE_HOME, ORACLE_SID, and CA_SCM_HOME.
   - Add $ORACLE_HOME/bin (for ORACLE) and $CA_SCM_HOME/bin to the PATH variable in the SCM UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   Note: $CA_SCM_HOME is the directory where the product's program files are installed on UNIX and Linux. (On Windows, this variable is named %CA_SCM_HOME%.)

   One method for performing both tasks is adding the following lines to the ~cascm/.profile file:

```
CA_SCM_HOME=/opt/CA/scm
```

   For Oracle, add the following lines:

```
ORACLE_HOME= /ora/app/oracle/product/9.2.0
ORACLE_SID=orcl

PATH=${CA_SCM_HOME}/bin:${ORACLE_HOME}/bin:${PATH}
export CA_SCM_HOME ORACLE_HOME ORACLE_SID PATH
```

   Note: The ORACLE_HOME=... line specifies a sample Oracle path name. If your Oracle path name is different, specify that path name instead of the sample path name. The ORACLE_SID=... line specifies a sample Oracle instance name. If your Oracle instance name is different, specify that instance name instead of the sample instance name.

The Public Key Infrastructure

You can install the Public Key Infrastructure (CAPKI) to help ensure the security of users, data, and applications in your enterprise.

CAPKI can be a shared component. The CAPKI setup behaves differently depending on if CAPKI is already installed or if it is a first-time install. If CAPKI was previously installed, the CAPKI setup runs again and it records an extra parent application (caller) to the CAPKI installation. In this case, you must run the setup program as the user who originally installed CAPKI, which is probably not the CASCM user.

Install CAPKI for All the Users on a Computer

You can install the Public Key Infrastructure (CAPKI) to help ensure the security of users, data, and applications in your enterprise.

Follow these steps:

1. Run the following commands:

   `cp /cdrom/ETPKI/etpki_platform.tar to /home/cascm/`

2. untar the etpki_*platform*.tar

   A folder structure is created. For example, on an AIX platform the directory structure is as follows:

   ```
   etpki_aix

     setup
     readme.txt
   ```

3. Change directories (cd) to the etpki_*platform* directory, and log in as the root user.

4. Run the following command:

   `setup install caller=CallerID options`

   You can specify the following options for this command:

   `CallerID`

   Specifies the parent application or component that is installing, and is dependent upon, CAPKI. This ID can be selected by users of CAPKI, so it is important that the ID uniquely identifies the parent product. When you have multiple subcomponents of a product and each component relies on CAPKI, use a CallerID that uniquely identifies each component. The maximum length of the identifier is 255 characters and it cannot contain spaces.

   CAPKI maintains a list of the CallerIDs of the products that installed it. When a product using CAPKI is uninstalled, the associated CallerID is removed from the list. And, when the list is empty, CAPKI is removed from the computer.

   For example, when installing CAPKI for the CA Harvest SCM server component, specify the callerID can be SCMSERVER. When installing the CA Harvest SCM client, specify callerID as SCMCLIENT. When installing the CA Harvest SCM agent, specify callerID as SCMAGENT.

`instdir=`*`path`*

Specifies an absolute path to the CAPKI installation directory. The installer determines the CAPKI installation directory that is based on the following factors in the given sequence:

a. Location that is specified by an existing CASHCOMP environment variable

b. Location that is specified by an existing CALIB environment variable (This path is done as previous versions of the CAPKI installer were dependent on CALIB)

c. Location specified in the instdir parameter

d. Default location: /opt/CA/SharedComponents/CAPKI

Note: CAPKI installation cause problems, if the required library libstdc++.so with version 5.0.2 is not found. Contact your administrator to install the required library.

`verbose`

Enables the output of diagnoses messages.

`env=<none|user|all>`

Sets environment variables for specific users. You can specify the following parameters:

`none`

(Default) Do not set environment variables.

`user`

Sets environment variables for *only* the current user ($HOME/.profile).

Installs to a custom location. It is mandatory to specify env=<user.

`all`

Sets environment variables for *all* users (/etc/profile).

CAPKI is installed on your computer, and if you specify to set environment variables, the following environment variables are set:

- CASHCOMP=Points to parent directory of the CAPKI install directory
- CALIB=Points to $CASHCOMP/lib
- CABIN=Points to $CASHCOMP/bin

Note: These variables are not set if env=none option is passed to the ETPKI r4.x (CAPKI) installer.

During the installation if you receive a return code of 0, CAPKI was successfully installed. If you receive a return code of 3, CAPKI did not install successfully. You can view a log file, capki_install.log in /tmp directory on non windows and in %TEMP% folder on windows machines. When you use the verbose option, the log file contains more messages.

Note: Previous versions of CAPKI used to set the ETPKIHOME variable. CAPKI (ETPKI 4.2.9) no longer sets or uses that variable.

Install CAPKI for a Particular User on a Computer

You can install the Public Key Infrastructure (CAPKI) for a particular user to help ensure the security of the user, data, and applications.

Follow these steps:

1. Log in to the computer as the user, for example, cascmuser.

   The install location is determined by criteria in the following in order:

   1. The location specified by an existing CASHCOMP environment variable
   2. The location specified by an existing CALIB environment variable (This is done as previous versions of the CAPKI installer were dependent on CALIB)
   3. A user specified location (see usage below for details)
   4. The default location: opt/CA/SharedComponents/CAPKI

2. To install in a location specified by the user, unset the CASHCOMP and CALIB variables in the environment if already defined:

   ```
   unset CASHCOMP

   unset CALIB
   ```

3. Run the following command:

   `setup install caller=CallerID options`

Example: Command to Install eTPKI for a Particular User on a Computer

```
setup install caller=SCMSERVER env=user instdir="/home/cascmuser" verbose
```

CallerID

Specifies the parent application or component that is installing, and is dependent upon, CAPKI. This ID can be selected by users of CAPKI, so it is important that the ID uniquely identifies the parent product. When you have multiple subcomponents of a product and each component relies on CAPKI, you must use a CallerID that uniquely identifies each component. The maximum length of the identifier is 255 characters and it cannot contain spaces.

CAPKI maintains a list of the CallerIDs of the products that installed it. When a product using CAPKI is uninstalled, the associated CallerID is removed from the list. And, when the list is empty, CAPKI is removed from the computer.

For example, when installing CAPKI for the CA Harvest SCM server component, specify the callerID can be SCMSERVER. When installing the CA Harvest SCM client, specify callerId as SCMCLIENT. When installing the CA Harvest SCM agent, specify callerID as SCMAGENT.

`instdir=path`

Specifies an absolute path to the CAPKI installation directory. By default, CAPKI is installed to /opt/CA/SharedComponents/CAPKI/lib.

Note: CAPKI installation may cause problems if the required library libstdc++.so with version 5.0.2 is not found. Contact your administrator to install the required library.

`verbose`

Enables the output of diagnoses messages.

`env=<none|user|all>`

Sets environment variables for specific users. You can specify the following:

`none`

(Default) Do not set environment variables.

`user`

Sets environment variables for *only* the current user ($HOME/.profile).

Installs to a custom location. It is mandatory to specify env=<user>

`all`

Sets environment variables for *all* users (/etc/profile).

CAPKI is installed on your computer, and if you specify to set environment variables, the following environment variables are set:

- CASHCOMP=Points to parent directory of CAPKI install directory
- CALIB=Points to $CASHCOMP/lib
- CABIN=Points to $CASHCOMP/bin

Note: These variables are not set if env=none option is passed to the ETPKI r4.x (CAPKI) installer.

If you receive a return code of 0, CAPKI was successfully installed. If you receive a return code of 3, CAPKI did not install successfully. You can view a log file, capki_install.log in /tmp directory on non-windows and %TEMP% directory on windows machines. When you use the verbose option, the log file contains additional messages.

Enable FIPS

You can enable (or disable) FIPS 140-2, which is an encryption standard that protects data from unauthorized programs and users. This option enforces compliance to FIPS 140-2, which has a specific set of standards for cryptographic modules.

Follow these steps:

1. Navigate to $CA_SCM_HOME and open the hserver.arg file.
2. Set FIPSMODE=1

   Note: The value 0 disables the FIPS mode.
3. Save and close the file.

   FIPS 140-2 is enabled.

Install CAI/PT ODBC

You can install CAI/PT ODBC on UNIX and Linux platforms supported by CA Harvest SCM.

If you are *upgrading* to the current release from a previous release such as 7.x, perform steps 1-7 to upgrade your existing CAI/PT ODBC.

Important! Before installing CAI/PT ODBC, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log on as the *cascm* user (the user named *cascm*).

Follow these steps:

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the CD mount point.
2. Copy the odbc311.tar.gz file to a temporary location. For example, to copy to the /tmp directory, enter the following command:

   ```
   cp /cdrom/bin/directory/odbc311.tar.gz /tmp
   ```

   In this command, *directory* sifies the directory for your UNIX or Linux platform.
3. Change to the temporary directory containing the odbc311.tar file and extract the contents. For example, enter the following commands:

   ```
   cd /tmp
   gunzip odbc311.tar.gz
   tar xvf odbc311.tar
   ```
4. Change to the odbc311_110106 directory and extract the CAI/PT ODBC install scripts from the tar file. For example, enter the following commands:

   ```
   cd /tmp/odbc311_110106
   tar xvf odbc311_UnixInstallScripts.tar
   ```
5. Enter the following command to run the shell script INSTALL.SH. This script configures the CAI/PT ODBC environment setup shell scripts odbcenv.sh, and odbcenv.csh.

   ```
   ./INSTALL.SH
   ```
6. The default CAI/PT ODBC installation directory is opt/CA/caiptodbc. Either accept the default or enter a different location for the installation directory.

   Note: If you are upgrading a previous CAI/PT ODBC installation and ODBC_HOME is not defined, enter the location of the target CAI/PT ODBC home (installation). If ODBC_HOME is defined and points to a valid CAI/PT ODBC installation, the ODBC_HOME target will be used.
7. Select the option for the database you are using.

   - If you are using Oracle, you are prompted to enter values for the ODBC database environment parameters. The product uses the Oracle environment parameters ORACLE_HOME and ORACLE_SID (1 and 2) *only*. If your ORACLE_HOME and ORACLE_SID settings are already defined, then the ODBC database environment parameters in the prompt are already set to match.
     - ORACLE_HOME specifies the home directory for the Oracle client installation.
     - ORACLE_SID specifies the local Oracle instance name. If the product server resides on the same network node as the Oracle database server, this value should be set. Setting this value enables the product to access a local database instance without using Oracle Net networking utilities.

   If the Oracle database server resides on a different network node than the product server, leave this value <not set>.

Note: If you are upgrading CAI/PT ODBC, the drivers are upgraded, and the upgrade is completed at this step.

8. Change to the ODBC installation directory, which is defined as ODBC_HOME by the ODBC setup scripts. For example, enter the following command:

   ```
   cd /opt/CA/caiptodbc
   ```

9. The CAI/PT ODBC environment shell scripts are now set up. Load the shell script odbcenv.sh or odbcenv.csh on your current shell to export the values set in the previous step.

   - If you are using the Bourne, Korn, or Bash shells, enter the following command:

     ```
     . ./odbcenv.sh
     ```

   - If you are using the C shell, enter the following command:

     ```
     source odbcenv.csh
     ```

   - Verify that ODBC_HOME is set by executing the echo command:

     ```
     echo $ODBC_HOME
     ```

Note: You might encounter ODBC installation failure message on the 64-bit Linux computers. However, the symptom and the solution are as follows:

Symptom:

While installing ODBC on Linux 64-bit computers, the installation fails with the following error message:

```
bin/installupd: error while loading shared libraries: libstdc++-libc6.2-2.so.3:
cannot open shared object file: No such file or directory PtODBC036: Error;
"bin/installupd" failed.
```

Solution:

Create a softlink for libstdc++-libc6.2-2.so.3 file with the latest available c++ standard library.

Install Enterprise Communicator (PEC)

You can install the Enterprise Communicator (PEC) on UNIX and Linux operating systems.

Important! Before installing PEC, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).
- Verify that the *cascm* user has write access to the /opt directory.

Follow these steps:

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the CD mount point.
2. Copy the PEC 474 file to a temporary location, for example, the /tmp directory, as follows:

   ```
   cp /cdrom/bin/directory/pec474.tar.gz /tmp
   ```

   Note: In this command, *directory* specifies the directory for your UNIX or Linux platform.
3. Change to the directory containing the PEC tar file. For example, enter the following command:

   ```
   cd /tmp
   ```
4. Extract the contents of the tar file in that directory, as follows:

   ```
   gunzip pec474.tar.gz
   tar xvf pec474.tar
   ```
5. Enter the following command to run the PEC installation script and make sure the configure_rtserver parameter is set to false, as follows:

   ```
   ./INSTALL.SH configure_rtserver=false
   ```
6. Either accept the default location (/opt/CA/pec) or enter a new location. The PEC files are installed to the directory you specified.
7. Add /opt/CA/pec to the PATH variable in the cascm UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   One method for performing both tasks is adding the following line to the ~harvest/.profile file:

   ```
   PATH=/opt/CA/pec:$PATH
   export PATH
   ```

Extract the Installation Files

Complete these steps whether you are performing a typical installation, a custom installation, or a first-time installation.

Important! Before starting these steps, do the following:

- On Linux, verify that the umask is set to 0022.
- On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Change to the installation directory. For example, if you are using the default installation directory, enter the following command:

   ```
   cd /opt/CA/scm
   ```

3. Enter the following command to copy the installation files for your UNIX or Linux platform to the current directory. Verify that you include the space and period at the end of the command; they represent the current directory.

   ```
   cp /cdrom/bin/directory/scm.tar.gz .
   ```

   directory

   Specifies the directory for your UNIX or Linux platform.

4. Enter the following commands to extract the files:

   ```
   gunzip scm.tar.gz
   tar xvf scm.tar
   ```

   The installation files are extracted.

Install the Server (Typical Installation)

Install the CA Harvest SCM server and select the *typical* installation type to:

- Install the product using predefined settings without providing any additional input.
- Install the server locally on SUSE LINUX 8.0 and 9.0 computers, the command-line utilities, and product documentation.

  Note: On UNIX and Linux, if you install the product server, the product client (command line utilities) and product agent are also installed automatically, whether the database is local or remote.

- Install these common, shared CA components: Enterprise Communicator (PEC), CA ODBC, eTPKI, and CA Licensing.
- Automatically use internal authentication for product users.

Note: If you want to install only the product client or agent, without the product server, install the server and use Oracle as your database, or determine the components to install (including external authentication such as OpenLDAP or Mixed Mode Authentication, and the CA Software Delivery integration), you should use a custom installation instead.

Before installing the server, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log in as the cascm user (the user named *cascm*).

To run the typical installation, enter the following commands.

```
cd /opt/CA/scm/install
./install.sh -noprompt
```

Install the Server (Custom Installation)

Install the CA Harvest SCM server and select the *custom* installation type to do the following:

- Configure settings in each step of the installation process.
- Use Oracle for your product database.
- Decide which components to install, including the server, command-line utilities, and product documentation.

  Note: On UNIX and Linux, if you install the server, the product client (command-line utilities) and product agent are also installed automatically, whether the database is local or remote.
- Install these common, shared CA components: Enterprise Communicator (PEC), CA ODBC, eTPKI, and CA Licensing
- Install CA Software Delivery.
- Enable FIPS mode for the product agent.
- Specify a firewall port range.
- Decide whether to use Internal, OpenLDAP, or Mixed Mode authentication for product users.

Note: If you want to install the server using predefined, default settings, you should use a typical installation instead.

Before installing the server, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- If you are using Oracle as your DBMS, log on as the *cascm* user.

During a first time CA Harvest SCM installation on UNIX or Linux, the installation script uses the Database Configuration Utility to create the required product tables in the database and load information about the product lifecycle templates into the tables.

Important! This procedure applies only if you are using a local database.

Follow these steps:

1. Change to the product installation directory. For example, enter the following command:

   ```
   cd /opt/CA/scm/install
   ```

2. Enter the following command to run the installation script:

   ```
   ./install.sh
   ```

3. The CA End User License Agreement (EULA) appears. Press the space bar to scroll down as you read the EULA. After reading the EULA, to accept the terms of the license agreement and continue with the product server installation, enter **Y** . If you do not accept the terms of the license agreement, enter **N** to stop the installation.

4. After you accept the licensing terms, the Third Party Software Acknowledgments appear. Press the space bar to scroll down and read the complete acknowledgment text. After reading the complete acknowledgment text, press Enter to continue with the installation.

5. From the installation options, select Option 1 to install the product server for the first time. This option also installs the CA Harvest SCM client (command-line utilities) and the CA Harvest SCM agent and prompts you to create a CA Harvest SCM database.

6. Specify whether to install the CA Software Delivery Integration.

   - If No, enter N and skip to the next step.
   - If Yes, enter Y and complete the related fields (see page 97). When the confirmation page appears, verify the values specified and continue.

7. Specify the method the product server and agent will use to authenticate users' logon credentials:

   Internal

   Uses internal (CA Harvest SCM) authentication. Login credentials provided to the broker are validated against the internal product user data.

   If you select Internal, skip the rest of this step and continue at the next step.

   OpenLDAP

   Uses an external server. Login credentials provided to the broker are validated against the external authentication server. If you select OpenLDAP authentication, you are prompted to supply the required LDAP-related information. These required LDAP values and optional values appear in confirmation pages so that you can verify them. Optional fields are automatically filled with default values. You can change them in the confirmation pages as needed.

   Note: When you install LDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

   Mixed Mode authentication

   Lets the SCMAdmin create users internally even though the authentication mode may be set to External (LDAP).

   Note: Mixed Mode authentication does not use LDAPserver for Authentication if users are created internally.

8. Specify the type of database.
9. The values for CA_SCM_HOME and DBMS-related environment variables appear. When prompted, select the appropriate number to modify a value; otherwise, accept the default [0] to continue.
10. If the eTrust Public Key Infrastructure (CAPKI) location is not available you may be prompted to enter the path to the CAPKI installation directory. If prompted, enter the complete path to the CAPKI installation directory. If not prompted, go to the next step.
11. If RTHOME is not set, you are prompted to enter the Enterprise Communicator (PEC) installation directory. If you are not prompted, the installation uses the current value of the environment variable, $RTHOME. If prompted, enter the Enterprise Communicator installation directory; the default is /opt/CA/pec.

12. To run the product server from behind a firewall, you must specify a range of available ports, as follows: At the Firewall port range prompt, specify the port range by entering the starting port number, a comma, and the ending port number. (Spaces are optional and are ignored.) For example, to specify a port range of 1500 through 1502, enter the following at the prompt:

    ```
    1500, 1502
    ```

    Important! The number of ports in the range must be greater than or equal to the maximum number of server processes and remote agents running behind the firewall.

13. The Database Configuration Utility starts. Use this utility to configure your database after you have installed your DBMS. You can optionally run the utility now or later, but you must run the utility before you can start using the product. For more information about this utility, see the The hdbsetup Database Configuration Utility (see page 223) section.

    Important! If you are using Oracle, do not use configdsn to set up your product database. You must use the Database Configuration Utility instead.

If any errors occurred during the installation script, see the $CA_SCM_HOME/install/log directory for log files containing information about the errors.

Note: The product server installation creates a log file named install.log for a first time installation or upgrade.log for an upgrade. The log file is in the $CA_SCM_HOME/install directory.

More information:

LDAP Compliant Directory Configuration Parameters (see page 101)

CA Software Delivery Integration Parameters

If you install CA Software Delivery (CA SDM), set the following parameters to configure the communication and the synchronization of data between CA Harvest SCM and CA SDM.

USD Server URL

Defines the network (IP) address or host name of the CA Harvest SCM server on which CA Harvest SCM will stage, create, and deploy CA SDM packages.

For a CA SDM server running on *Windows*, use the following format for the URL:

http://*server hostname or network address*/UDSM_R11_WebService/mod_gsoap.dll

For a CA SDM server running on *Linux*, use the following format for the URL:

http://*server hostname or network address*/UDSM_R11_WebService

Default: None

Limits: 255 characters

Note: The USD Server URL is stored in the usdserver=parameter in the \CA_SCM_HOME\HServer.arg file. For more information about viewing and updating the parameters in this file, see Set the URL for the CA Software Delivery Server (see page 322).

Example: Specify a Server URL Using the Server Name (Windows)

In this example, to specify the URL for a CA SDM server on Windows, using the server's name (usdsrv01), enter the following parameter:

http://usdsrv01/UDSM_R11_WebService/mod_gsoap.dll

Example: Specify a Server URL for a Network Address (Linux)

In this example, to specify the URL for a Linux server whose network address is 138.42.44.57, enter the following parameter:

http://138.42.44.57/UDSM_R11_WebService

Note: For more information about this URL format, see the information about the login web service in the *Unicenter Desktop and Server Management (DSM)* documentation.

USD Server User Name

Defines the user name for CA Harvest SCM to use to log on to the USD server.

Note: CA SDM is not required to be installed before the CA Harvest SCM server is installed. Therefore, the logon credentials for the USD server and CA Harvest SCM remote agent on the USD server are not validated at installation time. Instead, they are validated when you attempt to invoke one or more “USD” processes from CA Harvest SCM or to synchronize the CA SDM and CA Harvest SCM databases.

For a CA SDM server running on *Windows*, use the following format for the user name:

winnt://*login domain or host name*/*username*

For a CA SDM server running on Linux, use the following format for the user name:

unixl://*login domain or host name*/*username*

Default: None

Limits: 255 characters

Limits: 255 alphanumeric characters

Important! CA Harvest SCM automatically stores the USD server user name and password (if specified) in the \CA_SCM_HOME\husdsr.dfo file and encrypts this file, using the svrenc utility. To update the USD Server user name, password, or both, you must use the svrenc utility. For more information about using this utility to update them, see the *Command Line Reference Guide*.

Example: Specify the User Name Using the User's Domain (Windows)

In this example, to specify the user name (usdadmin) for a CA SDM server on Windows, using the name of the user's domain (domain01), enter the following parameter:

winnt://domain01/usdadmin

Example: Specify the User Name Using the Server Host Name (Linux)

In this example, to specify the user name (usdadmin) for a CA SDM server on Linux, using the host name (usdsrv01) of the CA SDM server, enter the following parameter:

unixl://usdsrv01/usdadmin

Note: For more information about this user format, see the information about the login web service in the *Unicenter Desktop and Server Management (DSM)* documentation.

USD Server Password

(Optional) Defines the password of the user specified in the USD Server User Name field. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

CA Harvest SCM Agent User Name

Defines the user name for CA Harvest SCM to use to access the CA Harvest SCM remote agent running on the USD server.

Note: CA SDM is not required to be installed before the CA Harvest SCM server is installed. Therefore, the logon credentials for the USD server and CA Harvest SCM remote agent on the USD server are not validated at installation time. Instead, they are validated when you attempt to invoke one or more "USD" processes from CA Harvest SCM or synchronize the CA SDM and CA Harvest SCM databases.

Default: None

Limits: 255 characters

Important! The product automatically stores the USD CA Harvest SCM agent user name and password (if specified) in the \CA_SCM_HOME\husdra.dfo file and encrypts this file, using the svrenc utility. To update the USD CA Harvest SCM Agent user name, password, or both, you must use the svrenc utility. For more information about using this utility, see the *Command Line Reference Guide*.

CA Harvest SCM Agent Password

(Optional) Defines the password of the user specified in the CA Harvest SCM Agent User Name field. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

CA Harvest SCM Agent Port

Defines the port for running the CA Harvest SCM agent.

Default: None

Minimum: 2

Maximum: 9999

Note: The agent does not run on port 1 or 1000.

Synchronization Interval Between SCM and USD Server

Defines the interval (in minutes) at which CA Harvest SCM synchronizes certain tables in the CA Harvest SCM database with those in the CA SDM database. CA Harvest SCM checks the contents of these tables in the CA SDM database and, if necessary, updates the CA Harvest SCM tables to match the CA SDM tables.

At this synchronization interval, CA Harvest SCM also queries for the status of any outstanding deployment jobs it has scheduled. For those jobs whose status has changed, the HARUSDPLATFORMINFO and HARUSDHISTORY tables are updated.

For more information about these CA Harvest SCM tables and others, contact Technical Support at http://ca.com/support.

Default: 60

Minimum: 15

Important! If you do not set this parameter, no synchronization occurs between CA Harvest SCM and the CA SDM server.

Limits: 6 digits

Note: The synchronization interval between CA Harvest SCM and the CA SDM server is stored in the usdsynchinterval= parameter in the \CA_SCM_HOME\HBroker.arg file.

LDAP Compliant Directory Configuration Parameters

Use the following LDAP settings to configure your CA Harvest SCM server to use LDAP Compliant Directory.

Note: The product installation program records your responses to the following prompts in the LDAP-related settings in the product configuration files HServer.arg, HAgent.arg, and HBroker.arg.

LDAP Server Name

Defines one or more host names of the LDAP server to which your CA Harvest SCM computer connects, for example:

```
hostname1
```

You can optionally define the port number to use on each host, by entering the host name in the form *hostname*:*port*, for example:

```
hostname2:389
```

You can specify a list of host names separated by spaces. Each host may optionally be of the form *hostname*:*port*, for example:

```
hostname1:389 hostname2 hostname3:389
```

Important! If used, the :port number specified in the LDAP Server Name field overrides the value specified in the LDAP Port Number field.

Limits: 255 characters

If the host name field defines multiple host names, the product computer connects to the first available LDAP server in the list.

LDAP Port Number

Specifies the port number for the LDAP server. This port number is used if the LDAP port number is not specified in the host name field.

Default: If you are using SSL as the encryption mechanism, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

Base Distinguished Name

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
"ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Search Filter

(Optional) Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to the product, this filter is used to search for the user in the LDAP server.

Default: (&(objectclass=person)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Examples

These examples use the default search filter.

If -ldapattrusrname=uid for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uid=<uid>))
```

If -ldapattrusrname=cn for your LDAP server, then the search filter is:

```
(&(objectclass=person)(cn=<cn>))
```

If -ldapattrusrname=uname for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uname=<uname>))
```

Examples: How the Search Filter is Used

The search filter is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

These examples use the default search filter and use the setting -ldapattrusrname=uid:

When the user amy33 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<amy33>))
```

When the user john22 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<john22>))
```

LDAP Search Timeout

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Default: 60

Limits: 20 digits.

Username Attribute ID

Defines your LDAP server's LDAP user attribute name for a user's user name.

Limits: 255 alphanumeric characters

LDAP/SASL Security/Encryption Mechanism

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security.

Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

None

Specifies no security mechanism.

Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

Default: None.

If you specify tls or ssl, complete the following fields; otherwise, skip them:

Trusted Certificate Filename

(Optional) Defines the complete path name of the TLS trusted certificate file.

This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (the product remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CACERT *filename*

Limits: 255 alphanumeric characters

Client Certificate Filename

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CERT *filename*

Limits: 255 alphanumeric characters

Client Key Filename

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the $HOME/.ldaprc file) using the following parameter:

TLS_KEY *filename*

Limits: 255 alphanumeric characters

Important! Private keys themselves are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

LDAP Distinguished Name

Defines the LDAP initial bind distinguished name (DN) to the LDAP server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
"cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Password for LDAP Distinguished Name

Defines the password for the LDAP distinguished name. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Your password is encrypted and is stored in the \CA_SCM_HOME\hserverauth.dfo file. This file name is specified in the following entry in the hserver.arg file:

```
ldapbindpwfile= hserverauth.dfo
```

Limits: 255 alphanumeric characters

Note: To update the encrypted password, use the command line utility svrenc. For more information about using svrenc, see the *Command Line Reference Guide*.

Fullname Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Full Name.

Limits: 255 alphanumeric characters

Phone Number Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Phone Number.

Limits: 255 alphanumeric characters

Phone Extension Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Phone Extension.

Limits: 255 alphanumeric characters

Fax Number Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for Fax Number.

Limits: 255 alphanumeric characters

E-mail Address Attribute ID

(Optional) Defines your LDAP server's LDAP user attribute name for eMail.

Limits: 255 alphanumeric characters

Synchronization frequency

Defines the authentication synchronization interval between the CA Harvest SCM broker and the authentication server. Use the input format *dd*[:*hh*[:*mm*[:*ss*]]], where *dd* is days, *hh* is hours, *mm* is minutes, and *ss* is seconds.

Default: 1 (1day)

Minimum: 0:1 (1hour)

Note: If the value of the authentication synchronization interval is invalid or less than one hour, the broker uses the minimum value (1 hour).

Limits: 20 characters

Examples:

-authsynchinterval=1:4 specifies 28 hours (1 day plus 4 hours).

-authsynchinterval=1:4:6 specifies 28 hours plus 6 minutes (1 day plus 4 hours plus 6 minutes).

-authsynchinterval=0:4:0:30 specifies 4 hours plus 30 seconds.

Note: When you install OpenLDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

UNIX ID Attribute

Defines your LDAP server's LDAP user attribute name for UNIX ID.

Limits: 255 alphanumeric characters

UNIX Group ID Attribute

Defines your LDAP server's LDAP user attribute name for UNIX Group ID.

Limits: 255 alphanumeric characters

UNIX Home Directory Attribute

Defines your LDAP server's LDAP user attribute name for UNIX Home Directory.

Limits: 255 alphanumeric characters

UNIX Shell Attribute

Defines your LDAP server's LDAP user attribute name for UNIX shell Directory.

Limits: 255 alphanumeric characters

You can optionally specify multiple base distinguished names when searching for user names in the LDAP server. To set up this capability, replace the existing description of the ldapbasedn=*base distinguished name* parameter with the following:

`ldapbasedn="name1[;name2[;name 3]…]"`

Defines one or more base distinguished names (DN) used when searching in the LDAP server.

To specify one base distinguished name, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com"
```

To specify two base distinguished names, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com;ou=europe,dc=abccorp,dc=com"
```

Important! When specifying multiple base distinguished names, separate them with a semicolon (;), as shown in the previous example.

Default: None

Limits: 255 characters

LDAP Parameters for External Usergroup Support

You can use the following parameters to define LDAP support for external user groups:

Note: Always enclose a value in quotation marks (" ") when it contains spaces.

-externalgroupenabled=1 or 0

(Optional) Use the following values to enable or disable your LDAP server's user group for external authentication:

- 1 enables your LDAP server's user group for external authentication.
- 0 disables your LDAP server's user group for external authentication.

-ldapgrpfilter=(&(objectclass=<objectclass of usergroup>)(usergroup-attribute-name=<placeholder>))

(Optional) Defines a group filter for locating a particular user group in the ldap server.

usergroup-attribute-name

Attribute of the usergroup used in -ldapattrusrgrpname=*attribute_name*.

<placeholder>

Identifies a literal constant placeholder for *usergroup-attribute-name*. Enter exactly the same value as *usergroup-attribute-name* and enclose the value with angle brackets (< >), as shown in the following example.

Example: Use -ldapattrusrgrpname=cn as the LDAP Server

In this example, if the objectclass of usergroup is group, and if -ldapattrusrgrpname=cn is used for your LDAP server, then the group filter is the following:

```
(&(objectclass=Group)(cn=<cn>))
```

-ldapattrusringrp=*attribute_name*

(Optional) Defines your LDAP server's attribute that evaluates members/users of a group, for example:

```
-ldapattrusringrp=member
```

-ldapattrgrpinusr=*attribute_name*

(Optional) Defines your LDAP server's attribute that evaluates groups of a user, for example:

```
-ldapattrgrpinusr=member_of
```

How to Configure the Server After Installation

Follow these steps:

1. If you need to create, update, or configure the database, do so now using the Database Configuration Utility.
2. If you need to change your configuration after you have completed your initial installation or upgrade, run the installation script again.
3. Set the license count.
4. (Optional) If you installed CA Software Delivery Integration, you must configure it before using it.
5. (Optional) If you installed OpenLDAP authentication, you must configure it before using it.
6. Start the broker.
7. Start the agent.

More information:

The hdbsetup Database Configuration Utility (see page 223)

Change the Configuration

If you need to change your CA Harvest SCM configuration after you have completed your initial product installation or upgrade, follow these steps; otherwise, skip these steps. Complete these steps *after* you complete any of the following steps:

- Change any settings for CA Software Delivery Integration, including enabling or disabling it.
- Change the product's authentication method or any of its settings.
- Update your ODBChome, RThome, or database path name.
- Change the database instance name (ORACLE_SID or II_INSTALLATION)

Note: If you have already deleted your installation files, repeat the steps in Extract the Installation Files (see page 93) before performing the following steps.

Follow these steps:

1. Navigate to the product installation directory. For example, enter the following command:

   ```
   cd  /opt/CA/scm/install
   ```

2. Enter the following command to run the installation script:
   ```
   ./install.sh
   ```

3. From the installation options, select option [3] - Change SCM installation configuration.

   This selection updates your installation to match the current user environment. If a DBMS environment variable is changed after the product is installed, this option is used to rebuild the command wrappers in CA_SCM_HOME/bin.

   You are prompted to enter the type of DBMS you are using.

4. If you are prompted for the installation directory of the eTrust Public Key Infrastructure (eTPKI), enter the complete path name.

5. If RTHOME has not been set as an environment variable, you are prompted for the Enterprise Communicator <RTHOME> installation directory. If prompted, enter the complete path name.
6. If the DBMS-related environment variables are not set, you are prompted to set them:
   - (Oracle). You may be prompted to set ORACLE_HOME and ORACLE_SID
   - If ODBC_HOME has not been set in the current environment, you are prompted to enter the installation directory for Open DataBase Connectivity directory (ODBC). If prompted, enter the complete path name. The default value for this path name (from the CAI/PT ODBC installation) is /opt/CA/caiptodbc). If you are not prompted, the installation program uses the current value of the $ODBC_HOME environment variable.

   The installation program creates the required product scripts, and displays a message similar to the following:

```
---------The database info has been successfully set---------
Installation completed
Completed time = <<< Mon Jan 28 14:34:05 EST 2005 >>>
Installation succeeded.
```

Set the License User Count

This procedure explains how to set the license user count for CA Harvest SCM.

Important! Before starting these steps, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).
- Verify that Lic98 has been installed on the product server. For Lic98 installation steps, see Install Lic98 Licensing.

If your site has obtained a product license certificate (an enterprise license key to be placed in a ca.olf file), you must set the product broker to the correct maximum license user count before starting any product component. If you have not received any licensing information, contact your account representative immediately to obtain a product license.

The default product server installation sets the user count to 50. In most cases, you must adjust this value to match the user count purchased for your site. The product defines one server process for each product user. Therefore, you must define the maximum license user count to match the maximum product server count.

The HBroker.arg file in the $CA_SCM_HOME directory defines the maximum product server count. A sample HBroker.arg file follows:

```
-minserver=5
-maxserver=50
-verbose
-queuesize=2
```

Verify that your HBroker.arg file and the -maxserver parameter setting match the maximum user count for your license: one product server process equals one product user. In this example, the -maxserver parameter is set to 10 product server processes, which equals the licensed maximum user count.

Note: If the value for -maxserver exceeds the licensed user count, you receive an error message similar to the following in the Lic98.log file for the product (typically found in the /opt/CA/ca_lic directory):

CA Licensing -2CHA - Usage limitation exceeded. Contact your account representative to obtain a new license.

If this message appears, either lower the -maxserver count or contact your account representative to obtain a new license.

More information:

Broker Options (UNIX, Linux, and zLinux) (see page 282)
Server Options (UNIX, Linux, and zLinux) (see page 285)

How to Configure the CA Software Delivery Integration (UNIX)

Note: Valid *only* if you installed the CA Software Delivery Integration.

If you installed CA SDM, before using it, verify that CA SDM is installed and configured in your environment.

More information:

External Authentication Configuration (see page 297)

OpenLDAP Authentication Configuration

Important! (Valid *only* if you installed OpenLDAP authentication.) Before you can use OpenLDAP authentication, you must configure the CA Harvest SCM components to use it.

The external authentication server should always have at least one user who is in the Administrator user group in CA Harvest SCM.

Note: The initial product user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of *1*. This user is always an administrator and always exists in the product, even if this user does not exist in the external authentication server. However, when you use external authentication, this user (like all other product users) must exist in the external authentication server to log in to the product.

Start the Broker

This procedure explains how to install the CA Harvest SCM broker.

Important! Before installing the product broker, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).

Important! On UNIX and Linux, when you start the product broker, the product server processes start automatically. Do not start the product server processes manually.

Follow these steps:

1. Change to the product's bin directory:

   ```
   cd $CA_SCM_HOME/bin
   ```

   For example, if the product is installed to the scm directory:

   ```
   cd /opt/CA/scm/bin
   ```

2. Enter the following command to start the product broker:

   ```
   ./bkrd
   ```

   When the product broker starts, the product server processes start automatically.

3. Enter the following command to verify that the product broker has started:

   ```
   ps -ef | grep bkrd
   ```

   A broker that has started successfully produces output similar to the following:

   ```
   scm 16329    1  1 13:31:47 ?    0:00    /opt/CA/scm/lib/bkrd
   scm 16371 16215  0 13:31:52 pts/1    0:00 grep bkrd
   ```

4. Enter the following command to verify that the product server processes have started:

   ```
   ps -ef | grep hserver
   ```

   A single server process that has started successfully produces output similar to the following:

   ```
   scm 16425 16329  0 13:32:17 ?   0:05 /opt/CA/scm/lib/hserver
   -parent=sparc -broker=sparc -version=default

   scm 16479 16215  0 13:39:35 pts/1    0:00 grep hserver
   ```

   Note: The default product installation will start five server processes.

   If either of the previous two commands does not return the expected number of processes, check for errors in the $CA_SCM_HOME/log directory (/scm/log in this example).

To send information to the standard output (screen) by starting the broker in verbose mode, enter the following command:

```
./bkrd -verbose
```

How to Start the Agent

The agent process on UNIX and Linux can be started in either multi-user mode or single-user mode for first-time installation and upgrade installations of the CA Harvest SCM server and agent. The multi-user mode agent requires system root privileges. The single-user mode agent can be started by a normal UNIX or Linux user account.

The user account starting a single-user agent process must have write access to the $CA_SCM_HOME/log directory; otherwise, a log file error similar to the following occurs:

```
Error: cannot open log file <Hagent_user> in <CA_SCM_HOME>
```

Note: For information about the single-user agent and its options, see the *Workbench User Guide*.

More information:

External Authentication Configuration (see page 297)

How to Prepare for the Client Command Line Utilities Installation

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the client until you have read that information and understand it.

   Note: You can find the Release Notes at http://ca.com/support.

2. Verify that installation directory paths have been set in your environment for the command line utilities, infrastructure (eTPKI), and Enterprise Communicator (PEC). If not, you must supply them during the installation.

3. Create the product user and default installation directories.

4. Install the Public Key Infrastructure (eTPKI) to help ensure the security of users, data, and applications in your enterprise.

5. Install Enterprise Communicator (PEC), a communications toolkit required for the product's client/server communication.

More information:

Create the CA Harvest SCM User and Default Directories (see page 115)
Install CAPKI for All the Users on a Computer (see page 85)
Install Enterprise Communicator (PEC) (see page 91)

Create the CA Harvest SCM User and Default Directories

Important! Before creating the CA Harvest SCM user and default directories, do the following:

- On Linux, verify that the umask is set to 0022.
- On UNIX, verify that the umask is set to 022.
- Log in as the root user (the user named root).

To create the product user and default directories on UNIX and Linux

1. Enter the following command to create the SCM group:

   ```
   groupadd cascm
   ```

2. Enter the following command to create a UNIX or Linux user named cascm who owns and runs the product server.

   ```
   useradd cascm
   ```

3. Enter the following command to add this user to the cascm group:

   ```
   usermod cascm -G cascm
   ```

4. Enter the following command to assign a password to this user.

   ```
   passwd cascm
   ```

   When prompted, specify the password.

5. If the product default directories do not exist, enter the following commands to create them:

   ```
   mkdir /opt/CA/scm
   mkdir /opt/CA/pec
   mkdir /opt/CA/ETPKI
   ```

   Enter the following commands to verify that the SCM user you created in Step 1 has write access to the required directories:

   ```
   chmod 775 /opt/CA/scm /opt/CA/ETPKI /opt/CA/pec
   ```

6. Enter the following commands to verify that the SCM group owns the following directories:

   ```
   chgrp -R cascm /opt/CA/scm
   chgrp -R cascm /opt/CA/ETPKI
   chgrp -R cascm /opt/CA/pec
   ```

7. Enter the following commands to verify that the SCM user owns the following directories:

   ```
   chown -R cascm /opt/CA/scm
   chown -R cascm /opt/CA/ETPKI
   chown -R cascm /opt/CA/pec
   ```

8. As the new SCM UNIX or Linux user (su cascm), complete these tasks:

   - Create the $CA_SCM_HOME environment variable.
   - Add $CA_SCM_HOME/bin to the PATH variable in the SCM UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   One method for performing both tasks is adding the following line to the ~cascm/.profile file:

   ```
   CA_SCM_HOME=/opt/CA/scm
   PATH=${CA_SCM_HOME}/bin:${PATH}
   export CA_SCM_HOME
   ```

Note: $CA_SCM_HOME is the directory where the program files are installed on UNIX and Linux. (On Windows, this variable is %CA_SCM_HOME%.)

Install CAPKI for All the Users on a Computer

You can install the Public Key Infrastructure (CAPKI) to help ensure the security of users, data, and applications in your enterprise.

Follow these steps:

1. Run the following commands:

   `cp /cdrom/ETPKI/etpki_platform.tar to /home/cascm/`

2. untar the etpki_*platform*.tar

   A folder structure is created. For example, on an AIX platform the directory structure is as follows:

   ```
   etpki_aix

     setup
     readme.txt
   ```

3. Change directories (cd) to the etpki_*platform* directory, and log in as the root user.

4. Run the following command:

   `setup install caller=CallerID options`

   You can specify the following options for this command:

   `CallerID`

   Specifies the parent application or component that is installing, and is dependent upon, CAPKI. This ID can be selected by users of CAPKI, so it is important that the ID uniquely identifies the parent product. When you have multiple subcomponents of a product and each component relies on CAPKI, use a CallerID that uniquely identifies each component. The maximum length of the identifier is 255 characters and it cannot contain spaces.

   CAPKI maintains a list of the CallerIDs of the products that installed it. When a product using CAPKI is uninstalled, the associated CallerID is removed from the list. And, when the list is empty, CAPKI is removed from the computer.

   For example, when installing CAPKI for the CA Harvest SCM server component, specify the callerID can be SCMSERVER. When installing the CA Harvest SCM client, specify callerID as SCMCLIENT. When installing the CA Harvest SCM agent, specify callerID as SCMAGENT.

`instdir=`*`path`*

Specifies an absolute path to the CAPKI installation directory. The installer determines the CAPKI installation directory that is based on the following factors in the given sequence:

a. Location that is specified by an existing CASHCOMP environment variable

b. Location that is specified by an existing CALIB environment variable (This path is done as previous versions of the CAPKI installer were dependent on CALIB)

c. Location specified in the instdir parameter

d. Default location: /opt/CA/SharedComponents/CAPKI

Note: CAPKI installation cause problems, if the required library libstdc++.so with version 5.0.2 is not found. Contact your administrator to install the required library.

`verbose`

Enables the output of diagnoses messages.

`env=<none|user|all>`

Sets environment variables for specific users. You can specify the following parameters:

`none`

(Default) Do not set environment variables.

`user`

Sets environment variables for *only* the current user ($HOME/.profile).

Installs to a custom location. It is mandatory to specify env=<user.

`all`

Sets environment variables for *all* users (/etc/profile).

CAPKI is installed on your computer, and if you specify to set environment variables, the following environment variables are set:

- CASHCOMP=Points to parent directory of the CAPKI install directory
- CALIB=Points to $CASHCOMP/lib
- CABIN=Points to $CASHCOMP/bin

Note: These variables are not set if env=none option is passed to the ETPKI r4.x (CAPKI) installer.

During the installation if you receive a return code of 0, CAPKI was successfully installed. If you receive a return code of 3, CAPKI did not install successfully. You can view a log file, capki_install.log in /tmp directory on non windows and in %TEMP% folder on windows machines. When you use the verbose option, the log file contains more messages.

Note: Previous versions of CAPKI used to set the ETPKIHOME variable. CAPKI (ETPKI 4.2.9) no longer sets or uses that variable.

Install Enterprise Communicator (PEC)

You can install the Enterprise Communicator (PEC) on UNIX and Linux operating systems.

Important! Before installing PEC, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).
- Verify that the *cascm* user has write access to the /opt directory.

Follow these steps:

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the CD mount point.
2. Copy the PEC 474 file to a temporary location, for example, the /tmp directory, as follows:

   ```
   cp /cdrom/bin/directory/pec474.tar.gz /tmp
   ```

   Note: In this command, *directory* specifies the directory for your UNIX or Linux platform.

3. Change to the directory containing the PEC tar file. For example, enter the following command:

   ```
   cd /tmp
   ```

4. Extract the contents of the tar file in that directory, as follows:

   ```
   gunzip pec474.tar.gz
   tar xvf pec474.tar
   ```

5. Enter the following command to run the PEC installation script and make sure the configure_rtserver parameter is set to false, as follows:

   ```
   ./INSTALL.SH configure_rtserver=false
   ```

6. Either accept the default location (/opt/CA/pec) or enter a new location. The PEC files are installed to the directory you specified.

7. Add /opt/CA/pec to the PATH variable in the cascm UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   One method for performing both tasks is adding the following line to the ~harvest/.profile file:

   ```
   PATH=/opt/CA/pec:$PATH
   export PATH
   ```

How to Prepare for the Client Components Installation

When you install the command-line utilities without the product server, the following client components are installed automatically:

- The product SDK components (HSDK and JHSDK).
- (On Linux) The Workbench-related files that let users launch the Workbench from the command line.

  Note: For information about launching the Workbench on Linux, see the *Workbench User Guide*.

Important! Before installing the command-line utilities, do the following:

- On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.
- Log on as the *cascm* user (the user named *cascm*).

Note: It is not necessary to install the CA Harvest SCM command-line utilities in an existing product server directory.

Install the Client Components

Installing the command-line utilities separately (without the product server) allows a product client to execute scripts and common processes from a UNIX or Linux shell. If you are installing the product *server* locally, the command-line utilities are included as part of the installation, so you do not need to install the command-line utilities separately.

Follow these steps:

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Change to the product installation directory. For example, if you are using the default product installation directory, enter the following command:

   ```
   cd /opt/CA/scm
   ```

3. To copy the command-line utilities installation files to the current directory, enter the following command. Verify that you enter the space and period at the end of the command; they represent the current directory.

   ```
   cp /cdrom/bin/directory/client.tar.gz .
   ```

   directory

   Specifies the directory for your UNIX or Linux platform.

4. To extract the command-line utilities files, enter the following commands:

   ```
   gunzip client.tar.gz
   tar xvf client.tar
   ```

5. Change to the product installation directory. For example, if you are using the default product installation directory, enter the following command:

   ```
   cd /opt/CA/scm/install
   ```

6. Run the setup.sh script. This script sets up the file structure necessary for executing the command-line utilities.

   ```
   ./setup.sh
   ```

7. After you execute ./setup.sh, the End User License Agreement (EULA) appears.

   Press the space bar to scroll down as you read the EULA. At the end of the EULA, accept the terms of the license agreement by using the Y key, and press Enter to continue with the command-line utilities installation. If you do not accept the terms of the license agreement, press the N key, and press Enter to halt the installation process.

8. After you accept the terms in the license agreement, the Third Party Software Acknowledgments appear. Press the space bar to scroll down and read the complete acknowledgment text. After reading the complete Acknowledgment text, press the Enter key to continue with the command-line utilities installation.

9. The script may prompt you for the location of the eTPKI or the PEC installation. Enter the paths that you used to install these components in the previous steps.

   The installation program creates the files and exits. The results of the installation are as follows:

   - A log file named clientinstall.log is located in the $CA_SCM_HOME/install directory.
   - The command-line utilities are installed.

     You can run the command lines from the bin directory, /opt/CA/scm/bin.
   - The product SDK components (HSDK and JHSDK) are installed.
   - (Linux) The Workbench-related files are installed.

How to Prepare for the Agent Installation

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the client until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.
2. Verify that installation directory paths have been set in your environment for the agent, eTrust Public Key Infrastructure (eTPKI), and Enterprise Communicator (PEC). If not, you must supply them during the installation.
3. Decide the method (either internal or OpenLDAP) the product should use to authenticate users' logon credentials.
4. Create the product user and default installation directories.
5. Install the Public Key Infrastructure (eTPKI) to help ensure the security of users, data, and applications in your enterprise.
6. Install Enterprise Communicator (PEC), a communications toolkit required for the product's client/server communication.
7. Extract the installation files.

More information:

Authentication Methods (see page 123)
Create the CA Harvest SCM User and Default Directories (see page 123)
Install CAPKI for All the Users on a Computer (see page 85)
Install the Enterprise Communicator (PEC) (see page 127)
Extract the Installation Files (see page 129)

Authentication Methods

During the CA Harvest SCM installation, you specify either *internal authentication* or *OpenLDAP authentication* as the method your site will use to authenticate users' names and passwords, for example, when a user attempts to log in to the product. Internal authentication uses the product to authenticate the user name and password, while OpenLDAP authentication uses an OpenLDAP authentication server for authentication.

The authentication method that you select may depend on your company's IT standards and conventions, resources, environment-specific concerns, manager input, and other factors in your company that influence IT-related decisions.

Consider all applicable factors, select your authentication method, and record your choice on the installation worksheet.

If you select *internal* authentication, you do not need to perform any preparation tasks.

If you select OpenLDAP authentication, prepare to install it.

For details about these TLS values, including how to specify them during the installation using either method, see the LDAP compliant directory configuration parameters (see page 101) for the TLS fields.

Create the CA Harvest SCM User and Default Directories

Important! Before creating the CA Harvest SCM user and default directories, do the following:

- On Linux, verify that the umask is set to 0022.
- On UNIX, verify that the umask is set to 022.
- Log in as the root user (the user named root).

To create the product user and default directories on UNIX and Linux

1. Enter the following command to create the SCM group:

   ```
   groupadd cascm
   ```

2. Enter the following command to create a UNIX or Linux user named cascm who owns and runs the product server.

   ```
   useradd cascm
   ```

3. Enter the following command to add this user to the cascm group:

   ```
   usermod cascm -G cascm
   ```

4. Enter the following command to assign a password to this user.

   ```
   passwd cascm
   ```

   When prompted, specify the password.

5. If the product default directories do not exist, enter the following commands to create them:

   ```
   mkdir /opt/CA/scm
   mkdir /opt/CA/pec
   mkdir /opt/CA/ETPKI
   ```

   Enter the following commands to verify that the SCM user you created in Step 1 has write access to the required directories:

   ```
   chmod 775 /opt/CA/scm /opt/CA/ETPKI /opt/CA/pec
   ```

6. Enter the following commands to verify that the SCM group owns the following directories:

   ```
   chgrp -R cascm /opt/CA/scm
   chgrp -R cascm /opt/CA/ETPKI
   chgrp -R cascm /opt/CA/pec
   ```

7. Enter the following commands to verify that the SCM user owns the following directories:

   ```
   chown -R cascm /opt/CA/scm
   chown -R cascm /opt/CA/ETPKI
   chown -R cascm /opt/CA/pec
   ```

8. As the new SCM UNIX or Linux user (su cascm), complete these tasks:

   - Create the $CA_SCM_HOME environment variable.
   - Add $CA_SCM_HOME/bin to the PATH variable in the SCM UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   One method for performing both tasks is adding the following line to the ~cascm/.profile file:

   ```
   CA_SCM_HOME=/opt/CA/scm
   PATH=${CA_SCM_HOME}/bin:${PATH}
   export CA_SCM_HOME
   ```

Note: $CA_SCM_HOME is the directory where the program files are installed on UNIX and Linux. (On Windows, this variable is %CA_SCM_HOME%.)

Install CAPKI for All the Users on a Computer

You can install the Public Key Infrastructure (CAPKI) to help ensure the security of users, data, and applications in your enterprise.

Follow these steps:

1. Run the following commands:

   `cp /cdrom/ETPKI/etpki_platform.tar to /home/cascm/`

2. untar the etpki_*platform*.tar

   A folder structure is created. For example, on an AIX platform the directory structure is as follows:

   ```
   etpki_aix

     setup
     readme.txt
   ```

3. Change directories (cd) to the etpki_*platform* directory, and log in as the root user.

4. Run the following command:

   `setup install caller=CallerID options`

   You can specify the following options for this command:

   `CallerID`

   Specifies the parent application or component that is installing, and is dependent upon, CAPKI. This ID can be selected by users of CAPKI, so it is important that the ID uniquely identifies the parent product. When you have multiple subcomponents of a product and each component relies on CAPKI, use a CallerID that uniquely identifies each component. The maximum length of the identifier is 255 characters and it cannot contain spaces.

   CAPKI maintains a list of the CallerIDs of the products that installed it. When a product using CAPKI is uninstalled, the associated CallerID is removed from the list. And, when the list is empty, CAPKI is removed from the computer.

   For example, when installing CAPKI for the CA Harvest SCM server component, specify the callerID can be SCMSERVER. When installing the CA Harvest SCM client, specify callerID as SCMCLIENT. When installing the CA Harvest SCM agent, specify callerID as SCMAGENT.

`instdir=path`

Specifies an absolute path to the CAPKI installation directory. The installer determines the CAPKI installation directory that is based on the following factors in the given sequence:

a. Location that is specified by an existing CASHCOMP environment variable

b. Location that is specified by an existing CALIB environment variable (This path is done as previous versions of the CAPKI installer were dependent on CALIB)

c. Location specified in the instdir parameter

d. Default location: /opt/CA/SharedComponents/CAPKI

Note: CAPKI installation cause problems, if the required library libstdc++.so with version 5.0.2 is not found. Contact your administrator to install the required library.

`verbose`

Enables the output of diagnoses messages.

`env=<none|user|all>`

Sets environment variables for specific users. You can specify the following parameters:

`none`

(Default) Do not set environment variables.

`user`

Sets environment variables for *only* the current user ($HOME/.profile).

Installs to a custom location. It is mandatory to specify env=<user.

`all`

Sets environment variables for *all* users (/etc/profile).

CAPKI is installed on your computer, and if you specify to set environment variables, the following environment variables are set:

- CASHCOMP=Points to parent directory of the CAPKI install directory
- CALIB=Points to $CASHCOMP/lib
- CABIN=Points to $CASHCOMP/bin

Note: These variables are not set if env=none option is passed to the ETPKI r4.x (CAPKI) installer.

During the installation if you receive a return code of 0, CAPKI was successfully installed. If you receive a return code of 3, CAPKI did not install successfully. You can view a log file, capki_install.log in /tmp directory on non windows and in %TEMP% folder on windows machines. When you use the verbose option, the log file contains more messages.

Note: Previous versions of CAPKI used to set the ETPKIHOME variable. CAPKI (ETPKI 4.2.9) no longer sets or uses that variable.

Install the Enterprise Communicator (PEC)

You can install the Enterprise Communicator (PEC) on UNIX and Linux platforms.

Important! Before installing PEC, do the following:

- On Linux, verify that the umask is set to 0022.
- On UNIX, verify that the umask is set to 022.
- Log in as the *cascm* user (the user named *cascm*).
- Verify that the *cascm* user has write access to the /opt directory.

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Copy the PEC 4.7 file to a temporary location, for example, the /tmp directory, as follows:

   ```
   cp /cdrom/bin/directory/pec474.tar.gz /tmp
   ```

   directory

   Specifies the directory for your UNIX or Linux platform.

3. Change to the directory containing the PEC tar file. For example, enter the following command:

   ```
   cd /tmp
   ```

4. Extract the contents of the tar file in that directory, as follows:

   ```
   gunzip pec474.tar.gz
   ```

   ```
   tar xvf pec474.tar
   ```

5. Enter the following command to run the PEC installation script and make sure the configure_rtserver parameter is set to false, as follows:

   ```
   ./INSTALL.SH configure_rtserver=false
   ```

6. Either accept the default location (/opt/CA/pec) or enter a new location. The PEC files are installed to the directory you specified.

7. Add /opt/CA/pec to the PATH variable in the product's UNIX or Linux user .profile [bash] or .cshrc [csh] file.

   One method for performing both tasks is adding the following line to the ~cascm/.profile file:

   ```
   PATH=/opt/CA/pec:$PATH
   export PATH
   ```

Extract the Installation Files

If you are performing an upgrade, verify that the CA Harvest SCM agent is not currently running before you attempt to install the new release. The following instructions apply to all supported UNIX and Linux operating systems.

Note: For the list of supported platforms, see the *Release Notes*.

1. Insert the installation media for your UNIX or Linux platform into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. To determine whether CA_SCM_HOME is set, enter the following command:

   ```
   echo $CA_SCM_HOME
   ```

   - If the $CA_SCM_HOME has been defined in the system variable, you must un-tar the agentonly.tar file into the $CA_SCM_HOME directory, using the following cd command:

     ```
     cd $CA_SCM_HOME
     ```

   - If the $CA_SCM_HOME has not been defined in the system variable, change to the installation directory. For example, if you are using the default installation directory, enter the following command:

     ```
     cd /opt/CA/scm
     ```

   Important! In this step, make sure that you use the correct cd command for your installation; otherwise, running the setup.sh script will not create scripts in the /bin directory.

3. To copy the agent installation files to the current directory, use the following command.

   Note: Verify that you include the space and period at the end of the command; they represent the current directory.

   ```
   cp /cdrom/bin/directory/agentonly.tar.gz .
   ```

   directory

   Specifies the directory for your UNIX or Linux platform.

4. To extract the agent files, use the following command:

   ```
   gunzip agentonly.tar.gz
   tar xvf agentonly.tar
   ```

   The installation files are extracted.

Install the Agent

The instructions in this procedure apply to agents on all supported UNIX and Linux platforms, including Novell SUSE LINUX for s/390. (In earlier releases, Novell SUSE LINUX for s/390 was documented separately.)

If you are installing the product *server* locally, the agent is included as part of the installation. It is not necessary to perform the agent installation.

The product agent only and command-line utilities only installations create a log file named clientinstall.log in the $CA_SCM_HOME/install directory.

On Linux, verify that the umask is set to 0022. On UNIX, verify that the umask is set to 022.

To install the CA Harvest SCM agent, extract the agent files and run the agent setup script. Detailed instructions follow. The setup.sh script sets up the file structure necessary for executing the agent.

1. Change to the product installation directory. For example, if you are using the default product installation directory, enter the following command:

   ```
   cd /opt/CA/scm/install
   ```

2. Enter the following command to run the setup.sh script:

   ```
   ./setup.sh
   ```

3. Specify the method the product server and agent will use to authenticate users' logon credentials: Internal or External.

   Internal

   Uses operating system calls. Login credentials provided to the remote agent are validated against the remote agent's operating system. If you select Internal, skip the rest of this step and continue at the next step.

   OpenLDAP

   This method uses an external server. Login credentials provided to the remote agent are validated against the external authentication server. If you select OpenLDAP authentication, you are prompted to supply the required LDAP-related information. For details about these fields, see LDAP Compliant Directory Configuration Windows.

   When you are finished specifying LDAP information, continue at the next step.

   Note: When you install LDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

4. Enter the agent port number.
5. The installation creates necessary files and exits.

LDAP Compliant Directory Configuration Parameters

Use the following settings to configure your CA Harvest SCM agent to use your LDAP Compliant Directory specifications.

Note: The product installation program records your responses to the following prompts described in the LDAP-related settings in the product configuration file HAgent.arg.

LDAP Server Name

Defines one or more host names of the LDAP server to which your CA Harvest SCM computer connects, for example:

```
hostname1
```

You can optionally define the port number to use on each host, by entering the host name in the form *hostname*:*port*, for example:

```
hostname2:389
```

You can specify a list of host names separated by spaces. Each host may optionally be of the form *hostname*:*port*, for example:

```
hostname1:389 hostname2 hostname3:389
```

Important! If used, the :port number specified in the LDAP Server Name field overrides the value specified in the LDAP Port Number field.

Limits: 255 characters

If the host name field defines multiple host names, the product computer connects to the first available LDAP server in the list.

LDAP Port Number

Specifies the port number for the LDAP server. This port number is used if the LDAP port number is not specified in the host name field.

Default: If you are using SSL as the encryption mechanism, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

Base Distinguished Name

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
"ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Search Filter

(Optional) Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to the product, this filter is used to search for the user in the LDAP server.

Default: (&(objectclass=person)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Examples

These examples use the default search filter.

If -ldapattrusrname=uid for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uid=<uid>))
```

If -ldapattrusrname=cn for your LDAP server, then the search filter is:

```
(&(objectclass=person)(cn=<cn>))
```

If -ldapattrusrname=uname for your LDAP server, then the search filter is:

```
(&(objectclass=person)(uname=<uname>))
```

Examples: How the Search Filter is Used

The search filter is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

These examples use the default search filter and use the setting -ldapattrusrname=uid:

When the user amy33 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<amy33>))
```

When the user john22 attempts to log on, the search filter used to locate this user is:

```
(&(objectclass=person)(uid=<john22>))
```

LDAP Search Timeout

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Default: 60

Limits: 20 digits.

Username Attribute ID

Defines your LDAP server's LDAP user attribute name for a user's user name.

Limits: 255 alphanumeric characters

LDAP/SASL Security/Encryption Mechanism

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security.

Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

None

Specifies no security mechanism.

Important! If you specify no encryption, user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear text.

Default: None.

If you specify tls or ssl, complete the following fields; otherwise, skip them:

Trusted Certificate Filename

(Optional) Defines the complete path name of the TLS trusted certificate file.

This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (the product remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CACERT *filename*

Limits: 255 alphanumeric characters

Client Certificate Filename

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CERT *filename*

Limits: 255 alphanumeric characters

Client Key Filename

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the $HOME/.ldaprc file) using the following parameter:

TLS_KEY *filename*

Limits: 255 alphanumeric characters

Important! Private keys themselves are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

LDAP Distinguished Name

Defines the LDAP initial bind distinguished name (DN) to the LDAP Server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
"cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"
```

Enter the quotation marks (" ") literally as shown.

Default: None

Limits: 255 characters

Password for LDAP Distinguished Name

Defines the password for the LDAP distinguished name. Do not enter spaces. If you do not specify a password, an empty password is used.

Limits: 255 alphanumeric characters

Your password is encrypted and is stored in the \CA_SCM_HOME\hagentauth.dfo file. This file name is specified in the following entry in the hagent.arg file:

ldapbindpwfile= hagentauth.dfo

UNIX ID Attribute

Defines your LDAP server's LDAP user attribute name for UNIX ID.

Limits: 255 alphanumeric characters

UNIX Group ID Attribute

Defines your LDAP server's LDAP user attribute name for UNIX Group ID.

Limits: 255 alphanumeric characters

UNIX Home Directory Attribute

Defines your LDAP server's LDAP user attribute name for UNIX Home Directory.

Limits: 255 alphanumeric characters

UNIX Shell Attribute

Defines your LDAP server's LDAP user attribute name for UNIX shell Directory.

Limits: 255 alphanumeric characters

You can optionally specify multiple base distinguished names when searching for user names in the LDAP server. To set up this capability, replace the existing description of the ldapbasedn=*base distinguished name* parameter with the following:

`ldapbasedn="name1[;name2[;name 3]…]"`

Defines one or more base distinguished names (DN) used when searching in the LDAP server.

To specify one base distinguished name, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com"
```

To specify two base distinguished names, use the format shown in the following example:

```
ldapbasedn="ou=america,dc=abccorp,dc=com;ou=europe,dc=abccorp,dc=com"
```

Important! When specifying multiple base distinguished names, separate them with a semicolon (;), as shown in the previous example.

Default: None

Limits: 255 characters

Start the Agent

This procedure applies to first-time installations of the CA Harvest SCM server and agent.

The agent process on UNIX and Linux can be started in either multi-user mode or single-user mode. The multi-user mode agent requires system root privileges. The single-user mode agent can be started by a normal UNIX or Linux user account.

The user account starting a single-user agent process must have write access to the $CA_SCM_HOME/log directory; otherwise, a log file error similar to the following occurs:

```
Error: can not open log file <Hagent_user> in <CA_SCM_HOME>
```

More information:

The Single-User Agent (see page 264)
External Authentication Configuration (see page 297)

OpenLDAP Authentication Configuration (Agent Installation UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Important! (Valid *only* if you installed OpenLDAP authentication.) Before you can use OpenLDAP authentication, you must configure the CA Harvest SCM components to use it.

The external authentication server should always have at least one user who is in the Administrator user group in CA Harvest SCM.

Note: The initial product user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of *1*. This user is always an administrator and always exists in the product, even if this user does not exist in the external authentication server. However, when you use external authentication, this user (like all other product users) must exist in the external authentication server to log in to the product.

Configure PAM Authentication for CA Harvest SCM Server and Agent

You can configure the CA Harvest SCM server and agent to use the Pluggable Authentication Module (PAM) configuration on UNIX platforms. You must implement and customize the PAM modules for the agent and server as per your business requirement.

Follow these steps:

1. Open the $CA_SCM_HOME/HAgent.arg file.
2. Modify the -authmode setting as follows:

   ```
   -authmode=pam
   ```

3. Save and close the file.

   The agent configuration file is saved with the new authentication mode.
4. Add "agntd" service to the PAM configuration of the computer.
5. Start the CA Harvest SCM agent as a root user.

   The agent is started with PAM authentication.

Follow these steps:

1. Open the $CA_SCM_HOME/HServer.arg file.
2. Modify the -authmode setting as follows:

   ```
   -authmode=pam
   ```

3. Save and close the file.

   The server configuration file is saved with the new authentication mode.
4. Add "hserver" service to the PAM configuration of the computer.
5. Start the CA Harvest SCM broker as a root user.

   The broker is started with PAM authentication.

Chapter 4: Installing on z/OS

Important! Installing these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

How to Prepare for the Agent Installation

To help ensure that you successfully install the CA Harvest SCM agent on z/OS, you must complete the following steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the agent until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Verify that you have a valid USS user ID to access the USS directories, a valid z/OS user, and that you are able to modify site-specific variables.

How to Install the z/OS Agent

Follow these steps:

1. Create the installation directories.
2. Extract the agent files.
3. Run the agent setup script.
4. Start the agent to verify that it has been installed successfully.

More information:

Create the Directories

The z/OS remote agent may be started in multi-user or single-user mode. Multi-user mode allows all users to log in to a single daemon process. The multi-user agent must be APF authorized and started by a user with root access. The multi-user agent impersonates users logged in to the daemon process.

The single-user agent may be started by a non-root user. The single-user agent supports login for the user ID that started the agent.

Follow these steps:

1. Verify that a z/OS user exists that owns and can execute the agent. This z/OS user requires the following:
   - Access to UNIX System Services (USS).
   - A valid OMVS segment in which a unique uid is associated with the z/OS login name.

   Note: If necessary, see your system or security administrator to create this user (see page 297).
2. Use the following commands to create a default destination directory for the components:

   ```
   mkdir /opt/ca
   ```

   ```
   mkdir /opt/ca/scm
   ```
3. Change the ownership of the directories to the product user created. For example, if the z/OS user ID is cascm, use the following command:

   ```
   chown -R cascm /opt/ca
   ```
4. Add $CA_SCM_HOME/bin to PATH in the .profile [bash] or .cshrc [csh] for the user. For example, if the user ID is cascm, add the following in ~cascm/.profile:

   ```
   CA_SCM_HOME=/opt/ca/scm
   PATH=${CA_SCM_HOME}/bin:${PATH}
   export CA_SCM_HOME PATH
   ```

More information:

The USS User ID (see page 276)

Extract the z/OS Agent Files

Verify that the z/OS agent is not currently running before you attempt to install the new release.

Follow these steps:

1. Navigate to the directory where you want to install the product agent with the following command:

   ```
   cd [install directory]
   ```

   In this command, for [*install directory*], substitute the path name of the directory you have selected. For example, to install the product agent in the CA Harvest SCM directory, enter the following:

   ```
   cd /opt/ca/scm
   ```

2. Binary FTP or copy the agent installation files from the installation media to the current directory, using one of the following commands:

   ```
   cp /cdrom/bin/zos/agentonly.tar .
   ```

   or

   ```
   ftp agentonly.tar .
   ```

   Note: Make sure to enter the space and a period at the end of the command. The space and period represent the current directory.

3. Extract the agent files using the following command:

   ```
   tar xvf agentonly.tar
   ```

Run the z/OS Agent Setup Script

The setup.sh script sets up the file structure necessary for executing the z/OS agent.

Follow these steps:

1. Navigate to the installation directory using the following command:

   ```
   cd install directory/install
   ```

   For example:

   ```
   cd /opt/ca/scm/install
   ```

2. Run the setup.sh script using the following command:

   ```
   ./setup.sh
   ```

3. Enter the port number for the agent.

   The installation creates the necessary agent files.

Start the z/OS Agent

To verify that you have successfully installed the z/OS agent, you should start the agent. To start the z/OS agent using the USS UNIX shell, use the following command:

```
./agntd –usr=username –pwd=password –port=1234
```

More information:

z/OS Agent Start Options (see page 273)
Starting and Stopping the z/OS Agent Using JCL (see page 142)

Stop the z/OS Agent

You can stop the z/OS agent using the USS UNIX shell at any time using the following command:

```
./agntd –port=1111 –usr=username -shutdown
```

More information:

z/OS Agent Start Options (see page 273)
Starting and Stopping the z/OS Agent Using JCL (see page 142)

Starting and Stopping the z/OS Agent Using JCL

As an alternative to the standard way to start and stop the z/OS agent (that is, using the *./agntd* command with options to start the agent and the *./agntd options -shutdown* command to stop the agent from the USS UNIX shell), you can write custom JCL to start the agent as a z/OS startup task, and write custom JCL to stop the agent. For example, you can use the IBM program BPXBATCH to run shell scripts or C executables (residing in HFS) from a TSO session or the z/OS console.

Example: Start the z/OS Agent with Custom JCL

In this example, you can use the following *sample*, customized JCL to start the z/OS agent.

```
S HAGENT
```

Note: This is sample JCL only. You may need to change it based on your requirements.

```
//HAGENT  PROC ACTION=''
//CASCM EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=6M,
//        PARM='SH /usr/lpp/cascm/bin/agntd &ACTION'
//STDENV  DD PATH='/usr/lpp/cascm/HAgent.arg'
//STDERR  DD PATH='/var/cascm/cascm.err',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRWXU)
//STDOUT  DD PATH='/var/cascm/cascm.out',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRWXU)
//SYSUDUMP DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

Example: Stop the z/OS Agent with Custom JCL

In this example, you can use the previously listed sample JCL, and add the following parameter to the JCL to stop the z/OS agent.

```
S HAGENT ACTION="-shutdown"
```

Special Codepage Translation

The SCM z/OS agent provides minimal support for codepage translation. By default, SCM translates text data from EBCDIC codepage 037 to ASCII codepage 437.

You can override the default codepage translation to specify EBCDIC codepage 1148 and ASCII codepage 1252; this special codepage translation override only applies to the z/OS agent and only applies to items stored in SCM that have been designated as text files (using the file extension) and are stored in a repository where items are compressed. All other text file items will be subject to the default codepage translation when transferring to and from the z/OS agent.

Note: For details about designating text files and configuring repositories for data compression, see the *Administrator Guide*.

To override the default codepage translation, insert the following parameters into HAgent.arg:

```
-agentcodepage=1148

-servercodepage=1252
```

Chapter 5: Installing the Client for the Mac OS

This section contains the following topics:

Install the Client Components

Installing the client allows a product client to execute scripts and common processes from a Mac shell.

Follow these steps:

1. Download the client.tar from ftp site to /Users/*currentuser*/Downloads.
2. Create a folder/directory for the CA Harvest SCM install.
3. Copy the client installation files to any directory where you want to install the client, and enter the following command using the Terminal utility:

   Note: Verify that you enter the space and period at the end of the command; they represent the current directory.

   For example:

   ```
   -cp /Users/currentuser/downloads/client.tar /Users/currentuser/CA/scm .
   ```

   Alternatively, you can move or copy client.tar to the installation location using Mac Finder.
4. To extract the CA Harvest SCM client binaries, enter the following commands:

   ```
   tar xvf client.tar
   ```

   Alternatively, you can extract all files in client.tar using the default Archive utility.
5. Change to the product installation directory. For example, if you are using the default product installation directory, enter the following command:

   ```
   -cd /Users/cascm/CA/scm/install
   ```
6. Run the setup.sh script. This script sets up the file structure necessary for executing the command-line utilities.

   ```
   ./setup.sh
   ```

   The End User License Agreement (EULA) appears.

7. Read and accept the EULA, and then the Third Party Software Acknowledgments. Press Enter.

   The installation continues with the command-line utilities installation. The installation program creates the files and exits. The results of the installation are as follows:

   - A log file named clientinstall.log is located in the *CA Harvest SCM install folder*/install directory.
   - The client binaries are installed.

8. (Optional) Double-click the workbench executable from the bin folder of the CA Harvest SCM install directory to launch the CA Harvest SCM workbench.

How to Prepare for the CA Harvest SCM Plug-In for Eclipse Installation on Mac OS

Follow these steps:

1. Install the CA Harvest SCM client.
2. Add the following variables to .profile for the current user to set up the environment variables:
   - CA_SCM_HOME=CA SCM client install folder (for example, /Users/cascm/CA/cascm)
   - PATH=$CA_SCM_HOME/bin:$PATH
   - DYLD_LIBRARY_PATH=$CA_SCM_HOME/lib:$DYLD_LIBRARY_PATH
   - CAPKI_HOME=$CA_SCM_HOME/lib
   - export CA_SCM_HOME PATH DYLD_LIBRARY_PATH CAPKI_HOME
3. Run the following command: Source .profile, and launch the Eclipse executable from the same terminal.

Java Runtime Requirement

CA Harvest SCM Plug-in for Eclipse supports the following versions of the Java runtime environment (JRE):

- Java 7 JRE (32-bit or 64-bit mixed mode)
- Java 6 JRE (32-bit or 64-bit mixed mode)

Chapter 6: Installing the Web Interface

Important! Installing the Web Interface is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

How to Prepare for the Web Interface Installation

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not install the Web Interface until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Verify that the CA Harvest SCM client has been installed. For your convenience, the Web Interface installation media includes the installation files for the product client on all supported platforms, including the product client (command-line utilities) for all supported UNIX and Linux platforms.

3. Verify that a supported web browser is installed on all client computers that will access the Web Interface.

   Note: For information about supported web browsers, see the *Release Notes*.

4. Verify that a supported release of the Java SDK Standard Edition is installed on your application server.

   Note: For information about supported releases of the Java SDK, see the *Release Notes*.

5. Verify that the CA Harvest SCM command-line utilities are installed on the same computer on which you will install the Web Interface.

6. Set and test system variables on your application server.

7. Install, start, or stop your application server software.

   Note: Stop the Web Application Server if it is Tomcat or Jboss. Start the Web Application Server if it is Sun Java System Web Server or Websphere. For information about supported application servers, see the *Release Notes*.

8. Verify that the product server is installed and running.

   Note: For information about how to start the server, see Start the Server (Windows) (see page 261) and Start the Server (UNIX and Linux) (see page 285).

9. Verify that the product database is installed and running.

10. If you are using *SQL Server*, review the SQL Server installation steps. In addition, find the SQL Server authentication user for database user.

11. If you are using *Oracle*, review the Oracle installation steps.

12. Verify the authentication mode by opening the HBroker.arg file on the CA Harvest SCM broker computer or the HServer.arg file on the product server computer and locate the -authmode=option. The product broker and server may be located either on the same computer or on different computers. In both files, the value is set to internal or external authentication (including mixed authentication mode) when the product server is installed. Record the value of the authentication mode.

13. Determine the CA Harvest SCM broker name.

14. Determine the deployment directory (that is, the full path to the web application server's home directory).

    Note: For WebSphere, determine the path name of the deployment node, which defines the path name of the WebSphere node on which you plan to deploy the Web Interface. For information, see WebSphere Node Path Names (see page 155).

15. Determine the context root, which is the name of the Web Interface instance.

16. For JBoss7 and later versions, the standalone option gets selected by default to deploy. For prior versions of JBoss, determine which JBoss configuration (either all or default) to deploy.

17. Determine whether you plan to authenticate usernames and passwords for the Web Interface using internal authentication or external authentication (OpenLDAP). You must specify the *same* authentication method for the Web Interface as the method the product server is using.
18. Determine if you want to install the Web Interface manually in unattended mode using a response file. If so, you must create a response file.
19. Determine if you want to deploy the Web Interface under your application server.
20. Determine the BusinessObjects URL. By default the BusinessObjects URL is populated with the following:

    `http://hostname:portnumber/InfoViewApp/logon.jsp`

    Change host_name and port_number to set your BusinessObjects report preference.

More information:

How to Set and Test System Variables (see page 149)
Find the SQL Server Authentication User for Database User (see page 152)
SQL Server Installation Steps (see page 153)
Oracle Installation Steps (see page 154)

How to Set and Test System Variables

Follow these steps:

1. Set the environment variables.
2. Set the system variables.
3. Set the system library path.
4. Test your changes.

More information:

Set the Environment Variables (Windows) (see page 150)
Set the System Variables (UNIX and Linux) (see page 150)
Set the System Library Path (UNIX and Linux) (see page 150)
Test Your Changes (see page 152)

Set the Environment Variables (Windows)

For all application servers running on Windows, set the following environment variables:

- **JAVA_HOME**-This system variable must point to the root directory of your Java SDK Standard Edition. For example:

  ```
  C:\jdk1.7.0_10
  ```

  Important! If you need to add this variable, enter it in your path *before* any other Java variable.

- **PATH**-Edit this environment variable to include the following item:

  ```
  %JAVA_HOME%\bin
  ```

Set the System Variables (UNIX and Linux)

For all application servers running on UNIX or Linux, set the following variables in the CA Harvest SCM user's profile:

JAVA_HOME

This system variable must point to the root directory of your Java SDK Standard Edition. For example:

```
/java1.7.10
```

Important! If you need to add this variable, enter it in your path before any other Java variable.

CA_SCM_HOME

This variable points to the directory where CA Harvest SCM is installed.

```
$CA_SCM_HOME
```

PATH

Edit this PATH variable to include the following items:

```
$JAVA_HOME/bin
$CA_SCM_HOME/bin
```

Set the System Library Path (UNIX and Linux)

Use the following examples to add the directories that CA Harvest SCM requires to the system library path. These examples are for the Bourne, Korn, or Bash shell.

UNIX

```
. /opt/CA/pec/bin/rtinit.sh
```

Solaris and Linux

```
LD_LIBRARY_PATH=$CA_SCM_HOME/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=app-server-path/web_interface-instance/
WEB-INF/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

where *app-server-path* is the deployment path for your application server, and *web_interface-instance* is the name of your Web Interface instance. For example:

```
LD_LIBRARY_PATH=/opt/Oracleiplanetwbsvr/http-U449742QA3.company.com/webapps/
http-U449742QA3.company.com/harweb/
WEB-INF/lib:$LD_LIBRARY_PATH
```

If you are deploying more than one instance of the Web Interface on your application server, you must include a library path statement for each instance. For example:

```
LD_LIBRARY_PATH=app-server-path/web_interface-instance-1/
WEB-INF/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=app-server-path/web_interface-instance-2/
WEB-INF/lib:$LD_LIBRARY_PATH
```

AIX

```
LIBPATH=$CA_SCM_HOME/lib:$LIBPATH
LIBPATH=app-server-path/web_interface-instance/WEB-INF/lib:$LIBPATH
export LIBPATH
```

where *app-server-path* is the deployment path for your application server, and *web_interface-instance* is the name of your Web Interface instance. For example:

```
LIBPATH=/usr/WebSphere/AppServer/installedApps/u449742qa2/harweb.ear/
harweb.war/WEB-INF/lib:$LIBPATH
```

If you are deploying more than one instance of the Web Interface on your application server, you must include a library path statement for each instance. For example:

```
LIBPATH=app-server-path/web_interface-instance-1/WEB-INF/lib:$LIBPATH
LIBPATH=app-server-path/web_interface-instance-2/WEB-INF/lib:$LIBPATH
```

HP-UX

```
SHLIB_PATH=$CA_SCM_HOME/lib:$SHLIB_PATH
SHLIB_PATH=app-server-path/web_interface-instance/WEB-INF/lib:$SHLIB_PATH
export SHLIB_PATH
```

where *app-server-path* is the deployment path for your application server, and *web_interface-instance* is the name of your Web Interface instance. For example:

```
/usr/jakarta-tomcat-5.5.4/webapps/web_interface/WEB-INF/lib
```

If you are deploying more than one instance of the Web Interface on your application server, you must include a library path statement for each instance. For example:

SHLIB_PATH=*app-server-path*/*web_interface-instance*-**1**/WEB-INF/lib:$SHLIB_PATH
SHLIB_PATH=*app-server-path*/*web_interface-instance*-**2**/WEB-INF/lib:$SHLIB_PATH

Test Your Changes

To test your changes, run the following command from the command prompt:

```
javac
```

If your environment is correctly configured for the Web Interface, you receive a message similar to the following, followed by a list of options:

```
Usage: javac options source_files
```

If you receive a message indicating that javac is not recognized, try the following:

- Set the variables.
- See your application server documentation.

Find the SQL Server Authentication User for Database User

During the Web Interface installation, you are prompted for the CA Harvest SCM database user name. Therefore, you must find and specify the database user name, or optionally create one if necessary.

If your DBMS is SQL Server, the Web Interface uses SQL Server authentication for its database connection, because the SQL Server JDBC driver supports SQL Server authentication *only*.

To find the database user, use the Database Configuration Utility (hdbsetup) to create a database user as an SQL Server authentication user.

More information:

The hdbsetup Database Configuration Utility (see page 223)

SQL Server Installation Steps

If you plan to use SQL Server, the CA Harvest SCM server and SQL Server must be running on a supported Windows operating system. However, product users on UNIX and Linux can use the Web Interface to connect to a product database running on SQL Server.

If you are using SQL Server, you should be familiar with the following SQL Server variables and record their values. These variables were set during the installation of the product server with SQL Server. You are prompted to specify these values during the Web Interface installation.

Record the values of these variables.

The *SQL Server JDBC driver version* is the JDBC driver you downloaded and installed.

The *SQL Server Authentication Mode* is the authentication mode that the Web Interface client computer uses to connect to the server.

The *SQL Server server name* is computer name or IP address of the computer on which SQL Server is installed and running.

Important! By default, TCP/IP is not enabled on SQL Server. You must enable TCP/IP, using the SQL Server Configuration Manager.

The *SQL Server instance name* is the name of the SQL Server instance. The default value is literally “default” (without quotation marks).

The *Database user name* is the user who has access to the product tables. This user is created during hdbsetup Create Repository or you can create it later with the Create Database User option.

The *Password* specifies the password of the database user.

The SQL Server database name is defined during hdbsetup Create SCM Repository.

Record the location of the JDBC driver files.

The installation program copies the required jar file or files to the directory named …\WEB-INF\lib (for Windows) or …/WEB-INF/lib (for UNIX and Linux).

Important! If the Web Interface deployment tool (the Web Interface GUI installation program) cannot find the required jar file or files at the location you specify, the tool continues to prompt you for a new location till one of the files is found. If you are using a Web Interface response file and a required jar file does not exist at the location you specify, the installation stops.

Download and install the appropriate JDBC driver for your environment from the Microsoft website. The SQL Server 2005 JDBC driver supports SQL Server 2005, and SQL Server 2008 JDBC driver supports SQL Server 2008.

Note: During the installation, if you select Microsoft SQL Server 2008 JDBC driver, it checks for both sqljdbc.jar and sqljdbc4.jar for copying to harweb.war/WEB-INF/lib folder.

- sqljdbc.jar class library requires a Java Runtime Environment (JRE) version 5.0. Using sqljdbc.jar on JRE 6.0 or JRE 7.0 throws an exception when connecting to a database.
- sqljdbc4.jar class library requires a JRE version 6.0 or 7.0. Using sqljdbc4.jar on JRE 5.0 throws an exception.

Based on the supported java version by the webserver, you can identify the unused jar file. After installing the harweb remove the unused jar file from the harweb/WEB-INF/lib folder.

Oracle Installation Steps

If are using Oracle, you should be familiar with the following Oracle variables and record their values. These variables are set during the Oracle installation.

Oracle host name

The computer name or IP address where the Oracle database is running. The Oracle hostname is *not* an Oracle TCP/IP service name.

Oracle port number

The Oracle listener port number. This is the TNS listener port number. The default is 1521 or 1526.

Oracle SID

The database instance identifier. The default is ORCL.

In addition, record the name and password of the Oracle user who owns the CA Harvest SCM tables. During the Web Interface installation, when you are prompted to connect from the Web Interface to Oracle, specify the user name and password of this user. Also note the temporary directory to be used by the installation (a default is provided by the wizard).

Note: For information about these variables, see your Oracle documentation.

Record the location of the appropriate JDBC driver files, for example:

- classes12.jar or ojdbc14.jar for Oracle10g.
- ojdbc5.jar or ojdbc6.jar for Oracle11g.

The installation program copies the .jar file to the directory named …\WEB-INF\lib (for Windows) or …/WEB-INF/lib (for UNIX and Linux).

If more than one .jar file is present in the location specified, then the installation program uses the appropriate one.

Important! If the Web Interface deployment tool (the Web Interface GUI installation program) cannot find one of these jar files at the location you specify, the tool continues to prompt you for a new location until one of the files is found. If you are using a Web Interface response file and neither jar file exists at the location you specify, the installation stops.

WebSphere Node Path Names

You can determine the correct value for the entry named Path Name of Deployment Node. In this entry, write the default or custom value for your version of WebSphere, as follows:

In *both* WebSphere 8.0 and 8.5, the default node is the short host name (for example, u449741aix1), not the fully qualified host name (for example, u449741aix1.ca.com).

In WebSphere 8.0 and 8.5, the default path name for the node is:

- On Windows: *WebSphere installation directory*\AppServer\profiles\AppSrv01\installedApps*node-name*
- On UNIX or Linux: *WebSphere installation directory*/AppServer/profiles/AppSrv01/installedApps/*node-name*

Example:

- On WebSphere 8.5, on Windows, if the node name is *computer-name*Node01Cell, the default path name for the node is *WebSphere installation directory*\AppServer\profiles\AppSrv01\installedApps*computer-name*Node01Cell.

Note: For more information about WebSphere nodes and their path names, see your WebSphere documentation.

Install the Web Interface (Windows)

Note: On the Web Interface installation media, the \Harweb directory contains the installation files for the Web Interface on all supported platforms.

Follow these steps:

1. Insert the Web Interface installation media in the drive of your Windows computer.
2. Open a command prompt and change directories to the Web Interface directory on the Web Interface installation media. For example:

   ```
   cd DVD-drive:\Harweb
   ```

3. At the command prompt, enter the following command to start the Web Interface Installation Wizard:

   ```
   HarwebInstall.bat
   ```

4. Follow the instructions on the wizard pages.

   Important! On the Web Interface Installation wizard, you can enter non-English characters (such as Japanese or accented characters) when you specify the name of the context root (.war file name). Note, however, that the .war file names with non-English characters may lead to problems due to system limitations in some servlet containers. If your .war filename contains non-English characters, Tomcat and other application servers may not be able to generate a context with a matching name. Therefore, you should use only single-byte, English characters when you name the .war file you are creating, especially for Tomcat.

5. When you are finished, click Finish.

   After you complete the installation procedure for your platform, the Web Interface Installation Wizard performs the following tasks upon exiting:

   - Creates a subdirectory named HARWEBHDT in the temporary directory you specified in the wizard
   - Creates the harweb.war file in the HARWEBHDT directory
   - Creates the harweb.rsp file in the temporary directory you specified in the wizard
   - Automatically deploys the harweb.war file to your application server, unless you selected the Package war file only check box
   - Creates or updates the harweb.cfg file and updates the web.xml file to work with your specific Web Interface installation
   - Copies the JDBC driver file or files required for your DBMS from the specified location to the ...\WEB-INF\lib directory (for Windows) or the .../WEB-INF/lib directory (for UNIX or Linux).

Install the Web Interface (UNIX, Linux, and zLinux)

Note:

- On the Web Interface installation media, the \Harweb directory contains the installation files for the Web Interface for all supported platforms.
- The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Follow these steps:

1. Insert the Web Interface installation media into the drive of your UNIX or Linux computer. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Change to the directory containing the Web Interface files.

   ```
   cd /cdrom/Harweb
   ```

3. Type the following command to start the Web Interface Installation Wizard:

   ```
   ./HarwebInstall.sh
   ```

   The wizard starts.
4. Follow the instructions on the wizard pages.

5. Click Finish.

   After you complete the installation procedure for your platform, the Web Interface Installation Wizard performs the following tasks upon exiting:

   - Creates a subdirectory named HARWEBHDT in the temporary directory you specified in the wizard
   - Creates the harweb.war file in the HARWEBHDT directory
   - Creates the harweb.rsp file in the temporary directory you specified in the wizard
   - Automatically deploys the harweb.war file to your application server, unless you selected the Package war file only check box
   - Creates or updates the harweb.cfg file and updates the web.xml file to work with your specific Web Interface installation
   - Copies the JDBC driver file or files that are required for your DBMS from the specified location to the …\WEB-INF\lib directory (for Windows) or the …/WEB-INF/lib directory (for UNIX or Linux).

Install Harweb for 64-bit WebSphere on AIX and Linux x86

For installing Harweb for 64-bit WebSphere on AIX or Linux x86, perform the following steps, after you install the Web Interface:

1. For AIX, copy the 64-bit library files from <HARWEB Installation folder>/WEB-INF/libaix64 to <HARWEB Installation folder>/WEB-INF/lib

   (or)

   For Linux x86, copy the 64-bit library files from <HARWEB Installation folder>/WEB-INF/liblinux64 to <HARWEB Installation folder>/WEB-INF/lib

2. Restart WebSphere.

Manually Install the Web Interface in Unattended Mode

You can install the Web Interface from the command line and not use the installation wizard.

Follow these steps:

1. Copy the following files from the Web Interface installation media to a temporary directory on your application server computer:
   - HarwebInstall.jar
   - harweb.war
2. Create a Web Interface response file (see page 160) containing your installation information, which the installer will read when performing the unattended installation.
3. Run one of the following commands from the command line, in the directory where HarwebInstall.jar and harweb.war are stored. These commands require that the database username and password are entered directly in the response file.

   Windows:

   ```
   java -Dcascmhome="%CA_SCM_HOME%" -jar HarwebInstall.jar -rsp path/harweb.rsp
   ```

   UNIX and Linux:

   ```
   java -Dcascmhome="$CA_SCM_HOME" -jar HarwebInstall.jar -rsp path/harweb.rsp
   ```

4. (Optional) For security reasons, if you do not want to enter the database username and password directly in the response file, you can use the following command to specify the user name and password on the command line instead.

   Windows:

   ```
   java -Dcascmhome="%CA_SCM_HOME%" -jar HarwebInstall.jar -rsp path\harweb.rsp
       -usr cascm_username -pwd cascm_password
   ```

   UNIX and Linux:

   ```
   java -Dcascmhome="$CA_SCM_HOME" -jar HarwebInstall.jar -rsp path/harweb.rsp
       -usr cascm_username -pwd cascm_password
   ```

   In this command, the -jar flag specifies the installation .jar file. The -rsp flag specifies your response file. If the response file is not in the same directory, specify the *path* to its location.

   When you enter the command, the installation program performs the following tasks:

   - Deploys the harweb.war file to your application server, unless you set the parameter WEBPACKAGEONLY=true. In this case, deploy the .war file manually.
   - Creates or updates the harweb.cfg file and updates the web.xml file to work with your specific Web Interface installation.

The Web Interface Response File

The Web Interface response file, harweb.rsp, is a text file created by the installation wizard. You can also create this file manually. The response file contains parameters you set using the required information during the installation. Create the response file for your installation using any text editor.

Example: Web Interface Response File (JBoss Application Server with an Oracle Database)

This example illustrates a Web Interface response file to install the Web Interface for the first time under JBoss Application Server using an Oracle database. In this example, comments are preceded with a hash mark (#), parameters appear in boldface, and parameters left blank do not apply to the installation. For example, if you are installing the Web Interface under Oracle Iplanet Web Server on UNIX or Linux, you must supply a domain name. Otherwise, you should leave the DOMAIN parameter blank. In addition, when specifying directory paths, you must use double backslashes (for example, TEMP=D\:\\temp).

Important! When you specify the name of the context root (the .war file name), you can enter non-English characters (such as Japanese or accented characters). If your .WAR filename contains non-English characters, Tomcat and other application servers may not be able to generate a context with a matching name. Therefore, you should use only single-byte, English characters when you name the .WAR file you are creating, especially for Tomcat.

```
#/n
#Wed Jan 30 14:59:54 GMT+05:30 2008
#Broker
BROKER=scm0321
#domain name (Example ca.com)
DOMAIN=
#Appserver Home directory
APPHOME=D\:\\jboss-4.2.2.GA
#Location of JDBC driver or drivers for Microsoft SQL Server
SJDBCLOC=
#Temp directory
TEMP=C\:\\Temp
#Context root
APPNAME=harweb
#Harweb Home Directory
HARWEBHOME=
#Jboss configuration (all, default). This value must be lowercase.
JBOSSCONFIGURATION=default
#SQL Server Authentication Mode (Windows/SQL Server)
SAUTHMODE=
#Oracle Port
OPORT=1521
```

```
#App Server Type (Tomcat 5.x, WebSphere Application Server, Sun Java System Web Server,
JBoss Application Server 4.x, JBoss Application Server 5.x)
APP_SERVER=JBoss Application Server
##Location of JDBC driver for Oracle
OJDBCLOC=C\:\\oracle\\product\\10.2.0\\db_1\\jdbc\\lib
#Oracle JDBC Driver version (10g or 11g)
OJDBCVERSION=10g
#Oracle Host Name
OHOSTNAME=mycomp-xp1
#Installation Type (New/Upgrade)
INSTALL_TYPE=New
#Package war file only (no deployment)
WEBPACKAGEONLY=false
#Database Password
DBPASSWORD=cascm
#Microsoft SQL Server JDBC Driver Version (2005)
SJDBCVERSION=
#Database User Name
DBUSER=smith156
# For WebSphere only-Path name of WebSphere node on which to deploy the Web Interface.
Defines the path name of the WebSphere node. For details about these path names, see
Path Names for WebSphere Nodes.
NODE_TO_DEPLOY_LOC=
#SQL Server Database Name
SDBNAME=
#Database Type (Oracle or SQLServer)
DBTYPE=Oracle
#CA SCM Authentication Mode (internal or openldap or mixed)
AUTHMODE=Internal
# SQL Server server name
SSERVERNAME=
#Oracle Service Name
OSERVICE=VM817
```

Deploy the Web Interface (Oracle Iplanet Web Server)

On Windows, UNIX and Linux, you can deploy the Web Interface under Oracle Iplanet Web Server; perform the following procedure from Step 2.

On Solaris, if you are not able to start Oracle Iplanet Server after deploying Web Interface with supported version of Oracle Iplanet Web Server following Step2, then perform Step1 and try to access Web Interface.

Follow these steps:

1. Update the Java Home by completing the following steps:
   a. Start the Administration Server.
   b. While running Oracle Iplanet Web Server, run the script set_start (this script was created during the Web Interface installation) from a console.
   c. Run the following command from same console.

      Replace *Oracle Iplanet Web Server Installation Home* with the installation home directory (for example, /bsohome/harweb/oracleiplanetwebserver) for Oracle iplanet web server.

      Replace *configuration node directory* with the configuration node directory name.

      `webservd -d Oracle Iplanet Web Server Installation Home/configuration node directory/config -r /b`

      For example:

      `webservd -d /bsohome/harweb/Oracle Iplanet/https-bso-sd-sun/config -r /b`

      Note: To stop the server instance, use the script (stopserv) provided by Oracle iplanet web server.
   d. Log in to the Admin console of Oracle iplanet web server.
   e. Select the specific configuration under Configuration Tasks.
   f. Click Edit Configuration.
   g. Select Java.
   h. In the Java Home path field, enter the full path to the JDK 1.6.x installation on the application server computer.
   i. Click Save and then deploy the pending changes.
2. Deploy the Web Interface by completing the following steps using the Admin Console of Oracle Iplanet Web Server:
   a. Start the Administration Server.
   b. Log in to the Admin console of Oracle Iplanet Web Server.
   c. Select the specific Virtual Server under Virtual Server Tasks.
   d. Click Add Web Application.
   e. Select the Specify a package file or a directory path that must be accessible from the server option.
   f. In the WAR file path field, enter the complete path name of the WAR file on the application server computer.
   g. For the Application URL, enter */webapp-name*

      Note: *webapp-name* is the name of your Web Interface application instance.

h. For Target Directory, select Default if you want to install under default path; otherwise, provide the specific path.

i. Click OK to deploy.

j. To apply the pending changes, click Deployment Pending.

k. Click Deploy.

l. Wait for it to finish.

Important! You can enter non-English characters (such as Japanese or accented characters) when you specify the name of the context root (WAR file name). Note, however, that .WAR file names with non-English characters may lead to problems due to system limitations in some servlet containers. If your .WAR filename contains non-English characters, Tomcat and other application servers may not be able to generate a context with a matching name. Therefore, you should use only single-byte, English characters when you name the .WAR file you are creating, especially for Tomcat.

m. (For Oracle Iplanet Server on Linux operating systems only) Copy the following libraries from $CA_SCM_HOME/lib to HARWEB_INSTALLATION_HOME/WEB-INF/lib:

```
libhcomm.so
libhutils.so
libharagent.so
libsignfile.so
libhauth.so
libhauthserver.so
```

n. Restart the Oracle Iplanet Web Server.

o. Point a browser to http://*host-name*[:*port*]/*webapp-name*/

Note: *host-name* is the name of your application server host computer, :*port* is an optional port number, and *webapp-name* is the name of your Web Interface application instance.

p. When the initial CA Harvest SCM page appears, verify that you can log in to the Web Interface.

Deploy the Web Interface (Apache Tomcat)

On Windows, UNIX, and Linux, you can deploy the Web Interface under Apache Tomcat.

Follow these steps:

1. Stop Apache Tomcat.
2. Start Apache Tomcat.
3. Open an internet browser and go to http://*host-name[:port]*/*webapp-name*/

   host-name is the name of your application server host computer, *:port* is an optional port number, and *webapp-name* is the name of your Web Interface application instance.
4. When the initial CA Harvest SCM page appears, verify that you can log in to the Web Interface.

Deploy the Web Interface on WebSphere (Windows)

You can deploy the Web Interface on WebSphere for Windows.

Follow these steps:

1. Verify that the target WebSphere Application Server is running.
2. Start the WebSphere Application Server Administrative Console.
3. Expand the Applications entry in the Actions tree in the left frame.
4. Click Install New Application.
5. In the Preparing for the Application Installation dialog, under Path, select Local Path and enter the path name for the fully qualified path to the harweb.war file created when you installed the Web Interface.

   Important! You can enter non-English characters (such as Japanese or accented characters) when you specify the name of the context root (WAR file name). Note, however, that .WAR file names with non-English characters may lead to problems due to system limitations in some servlet containers. If your .WAR filename contains non-English characters, Tomcat and other application servers may not be able to generate a context with a matching name. Therefore, you should use only single-byte, English characters when you name the .WAR file you are creating, especially for Tomcat.
6. Under Context Root, enter /*webapp-name*, where *webapp-name* is the name of your web application, and click Next.

7. Set the following options:
   - Activate the Generate Default Bindings check box.
   - Click the button named Do not override existing binding.
   - Click the button named Default virtual host name for web modules.
8. Click Next.
9. In the Application Security Warning dialog, click Continue.
10. For Step 1 in the Install New Application dialog, enter an appropriate name and specify the Web Interface as the application name. The default name is harweb.war. Click Next.
11. Check harweb.war with virtual host mapping as the default_host. Click Next.
12. Select appropriate site-specific options. Click Finish when completed.
13. On the Enterprise Applications frame, select the check box for the name you previously gave this application. Click Next.
14. Click Finish and Save the configuration.
15. Select the application and click Start.

    The Web Interface is now deployed into your application server and is addressable as */webapp-name*, where *webapp-name* is the name of your Web Interface application instance appended to your web application server's URL.

Deploy the Web Interface on WebSphere (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can deploy the Web Interface on WebSphere for UNIX and Linux.

Follow these steps:

1. Verify that the target WebSphere Application Server is running.
2. Start the WebSphere Application Server Administrative Console.
3. Expand the Applications entry in the Actions tree in the left frame.
4. Click Install New Application.

5. In the Preparing for the Application Installation dialog, under Path, select Local Path and enter the path name for the fully qualified path to the harweb.war file created when you installed the Web Interface.

   Important! You can enter non-English characters (such as Japanese or accented characters) when you specify the name of the context root (WAR file name). Note, however, that .WAR file names with non-English characters may lead to problems due to system limitations in some servlet containers. If your .WAR filename contains non-English characters, Tomcat and other application servers may not be able to generate a context with a matching name. Therefore, you should use only single-byte, English characters when you name the .WAR file you are creating, especially for Tomcat.

6. Under Context Root, enter */webapp-name*, where *webapp-name* is the name of your Web Interface application instance and click Next.
7. Activate the Generate Default Bindings check box. Click Next.
8. In the Application Security Warning dialog, click Continue.
9. In Step 1 in the Install New Application dialog, enter an appropriate name and provide the Web Interface as the entry. Click Next.
10. Select appropriate site-specific options. Click Finish when completed.
11. On the Enterprise Applications frame, select the check box for the name you previously gave this application.
12. Click Start.

    The Web Interface is now deployed into your application server and is addressable as */webapp-name* appended to your web application server's URL.

Deploy the Web Interface (JBoss)

You can deploy the Web Interface under JBoss for Windows, UNIX, and Linux.

Note: JBoss7 and later versions are supported only with Java 6 and later versions.

Follow these steps:

1. (Optional) Stop JBoss.
2. Complete one of the following steps:
   - Copy the harweb.war file to the directory from which you want to start JBoss.

     Windows

     - If you specified during installation to deploy all JBoss services (JBOSSCONFIGURATION=all), copy harweb.war to %JBOSS_HOME%\server\all\deploy.
     - If you specified during installation to deploy the default JBoss services only (JBOSSCONFIGURATION=default), copy harweb.war to %JBOSS_HOME%\server\default\deploy.If you specified during installation to deploy the standalone JBoss services only (JBOSSCONFIGURATION=standalone), copy harweb.war to %JBOSS_HOME%\standalone\deployments.

     UNIX and Linux

     - If you specified during installation to deploy all JBoss services (JBOSSCONFIGURATION=all), copy harweb.war to $JBOSS_HOME/server/all/deploy.
     - If you specified during installation to deploy the default JBoss services only (JBOSSCONFIGURATION=default), copy harweb.war to $JBOSS_HOME/server/default/deploy.
     - If you specified during installation to deploy the standalone JBoss services only (JBOSSCONFIGURATION=standalone), copy harweb.war to %JBOSS_HOME%/standalone/deployments.

   Note: Steps 3 and 4 are not applicable if the JBoss version is 7 or later.
3. Create a directory named harweb.war and expand the harweb.war file to that directory, as follows:
   - (Windows) Create the harweb.war directory under either *%JBOSS_HOME%**server**default*\deploy or %JBOSS_HOME%\server*all*\deploy, whichever directory matches your selection at installation time.
   - (UNIX and Linux) Create the harweb.war directory under either *$JBOSS_HOME/server/default*/deploy or $JBOSS_HOME/server/*all*/deploy, whichever directory matches your selection at installation time.

4. Go to the lib folder and rename the jar file as follows:
   - (Windows) Go to %JBOSS_HOME%\server\default\deploy\harweb.war\WEB-INF\lib or %JBOSS_HOME%\server\all\deploy\harweb.war\WEB-INF\lib folder and rename log4j.jar to log 4j.jar_old.
   - (Linux/Unix) Go to either $JBOSS_HOME/server/default/deploy/harweb.war/WEB-INF/lib or $JBOSS_HOME/server/all/deploy/harweb.war/WEB-INF/lib folder and rename log4.jar to Log4j.jar_old
5. If you stopped JBoss, restart it.
6. Open a browser and access http://*host-name[:port]*/*webapp-name*/

   host-name is the name of your application server host computer, *:port* is an optional port number, and *webapp-name* is the name of your Web Interface application instance.

   When the initial CA Harvest SCM page is displayed, verify that you can log in to the Web Interface.

How to Complete the Web Interface Installation

Follow these steps:

1. Start the Web Interface.
2. Configure the Web Interface settings.
3. Convert custom form types.
4. Customize the Web Interface form type search.
5. Implement lifecycle diagrams.
6. (Optional) If you are using HTTPS, set up the Web Interface to use this secure protocol.
7. (Optional) If your installation is an international (non-English) Web Interface installation, configure the Web Interface for international use.

More information:

Start the Web Interface Instance (see page 169)
Web Interface Configuration Settings (see page 169)
Custom Form Types (see page 172)
Web Interface Form Type Search (see page 172)
Set Up the Web Interface to Use HTTPS (see page 172)
Configure the Web Interface for International Languages (see page 172)

Start the Web Interface Instance

Start the Web Interface from the internet browser using the following URL:

`http://host-name/webapp-name`

host-name

Specifies the name of your application server host computer.

webapp-name

Specifies the name of your Web Interface application instance.

If your application server requires a port number in the URL, add the **:***port* number after the server name in the URL. For example:

`http://host-name:port/webapp-name`

See your application server documentation to determine if your application server requires a port number in the URL.

The URL is case sensitive.

Apache Tomcat uses the default port of 8080. Check your application server settings for the correct default port number.

Important! Do not use the same port number for the Apache Tomcat and Oracle (or SQL Server) database servers. By default, the Apache Tomcat server uses port 8080, and if the Oracle (or SQL Server) database also uses the same port, then an error results.

Web Interface Configuration Settings

When the Web Interface is installed, a configuration file named harweb.cfg is created in the \harweb\WEB-INF directory. The following settings can be modified:

Authentication mode

Specifies whether to use internal (CA Harvest SCM) authentication or external authentication, such as OpenLDAP authentication.

`AuthMode=internal|openldap`

Sets the Web Interface authentication mode to the *same* value used by the Web Interface broker. You typically set the authentication mode during the Web Interface installation. If you change the authentication mode for the Web Interface broker after the Web Interface installation, change the authentication mode for the Web Interface to the same value as its broker. For example, if you change the Web Interface broker to use OpenLDAP authentication after the installation, update the harweb.cfg file to specify the setting AuthMode=openldap, which configures the Web Interface to use OpenLDAP authentication (see page 49).

`MixedAuthMode={1|0}`

(Optional) Specifies whether the CA Harvest SCM server uses mixed-mode authentication.

1

Allows internal (CA Harvest SCM) accounts to be maintained and authenticated.

0

Specifies all accounts to be external.

Default: N

Broker

Specifies the product broker name.

Rtserver

Specifies the product RTServer name.

Rtserverport

Specifies the product RTServer port.

Note: For an RTServer running on nondefault port, modify rtserverport with the appropriate port number.

Working Directory

An administrator should manage this directory for space considerations.

- (Windows) The path delimiter must be a double backslash. The default is install-dir\\harweb\\temp.
- (UNIX and Linux) The default working directory is install-dir/harweb/temp.

JDBC Driver Name

Specifies the name of the JDBC driver that the Web Interface uses to register the database driver before connecting to the product database. Use the following information to specify the JDBC driver for the DBMS.

DBMS	JDBC Driver
Oracle	oracle.jdbc.driver.OracleDriver
SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver

Database Connection String

Specifies the settings that the Web Interface uses to connect to the product database. If you change any values used in the database connection string, record the new values and edit the database connection string in the Web Interface configuration file (harweb.cfg) to specify the new values. You must specify these values to help ensure that the product and the Web Interface continue to connect to the same database.

Example: Database Connection String (Oracle)

This example illustrates the database connection string for Oracle.

```
JDBCConnectionURL=jdbc:oracle:thin:@[hostname]:[port-number]:[sid]
```

hostname

Specifies the computer name or IP address where the Oracle database is running. The Oracle hostname is *not* an Oracle TCP/IP service name.

port-number

Specifies the Oracle listener port number. This value is the TNS listener port number. The default is 1521 or 1526.

sid

Identifies the database instance. The default value is ORCL.

The following is a sample Oracle database connection string.

```
JDBCConnectionURL=jdbc:oracle:thin:@santbarb110:1526:ORCL
```

Example: Database Connection String (SQL Server 2005)

This example illustrates the database connection string for SQL Server 2005 JDBC Driver.

```
JDBCConnectionURL= jdbc:sqlserver://[servername];instanceName=[instancename];
databaseName=[ databasename];forwardReadOnlyMethod=serverCursor
```

servername

Specifies the computer name or IP address where the SQL Server database is running. The *port-number* is the SQL Server listener port number. The default is 1433.

databasename

Specifies the name of the SQL Server database.

instancename

(SQL Server 2005 JDBC Driver) Specifies the instance name of your SQL Server. The default instance name is literally “default” (without quotation marks). If you are using default instance, you do not need to specify the *instancename* in the connection string.

The following is a sample SQL Server 2005 JDBC driver database connection string.

```
JDBCConnectionURL=
jdbc:sqlserver://santbarb110;instanceName=v2005;databaseName=mdb;forwardReadOnlyM
ethod=serverCursor
```

Custom Form Types

If you are using custom forms, convert them for use in the current release of the product.

More information:

How to Convert Customized Form Types and Add Them to the Database (see page 211)

Web Interface Form Type Search

The Web Interface Find Form page lets you execute filtering operations to locate forms with common attributes.

Set Up the Web Interface to Use HTTPS

If you are using HTTPS, when you set up the Web Interface to be served under an HTTPS server, the entire content hierarchy of the Web Interface is served through this protocol.

The exact procedures you use to set up HTTPS depend on your application server configuration and are independent of the Web Interface. Refer to the documentation for the application server to configure the server to use HTTPS. After the application server is set up to use HTTPS, the Web Interface is set up to be served under an HTTPS server.

Configure the Web Interface for International Languages

Important! This optional step is only for non-English Web Interface installations.

For more information about configuring the Web Interface for a non-English installation, see International Language, Character Set, and Locale Settings.

Note: After installing the Web Interface, do *not* update the default ISO-8859-1 encoding in the ../harweb/WEB_INF/web.xml file.

Web Interface Configuration Errors

The following actions help you correct common configuration errors that you may experience with the Web Interface.

HTTP Status 404 Errors

HTTP Status 404 errors indicate that the web.xml file is incorrect. To fix the error, rename one of the other web*.xml files to web.xml.

HTTP Status 500 Errors

HTTP Status 500 errors indicate a problem in the harweb.cfg file.

The following are possible causes:

- One or more of the following is invalid: the user name, password, hostname, port number, database ID, or working directory.
- The JDBC driver or Webext is missing.
- For Oracle, check the Oracle Listener and start it if necessary.
- The Webext does not load; this problem occurs if more than one instance of the Web Interface is being used with the DFO option.

HTTP Status -1000 Errors

HTTP Status -1000 errors indicate one of the following problems:

- The command line is missing or not in the operating system variable path name variable. Check the variable and correct it if necessary.
- The command line parsing has been broken. To see if this problem exists, enter several commands with different options at the command line. Enter each command twice: once as you would typically enter the command and once with the -di option. For example, enter the following hchu commands:

  ```
  hchu input.txt
  hchu -di input.txt
  ```

  For both commands in this example, the input.txt is:

  ```
  -b broker -usr username -pw password -l
  ```

If you are using Tomcat, start the command line as the same user that starts Tomcat.

Java Errors

Java errors indicate a problem with Java. Possible causes are that *only* the Java Runtime Environment (JRE) is installed or that CA Harvest SCM does not support the Java SDK Standard edition installed. If either condition is true, the new Web Interface installation fails and custom forms do not compile.

Note: For the versions of Java SDK Standard edition that the product supports, see the Release Notes.

Web Interface requires a larger heap space when working with large numbers of versions. If you receive a "java.lang.OutOfMemoryError: Java heap space" error, the Java Maximum Heap Size needs to be increased for the Application Server/Servlet Engine. The option that your Application Server/Servlet Engine will need to pass to Java is –Xmx. For example, to set the Maximum Heap Size to 512 MB, change the –Xmx option to –Xmx512m. Refer to the Application Server/Servlet Engine documentation for instructions on completing this task.

Form Errors

Form errors indicate problems with generating custom forms. If you experience form errors, verify that a version of the Java SDK Standard edition that the product supports is installed and is defined in the operating system PATH variable.

Verify that you have set and tested all applicable system variables. For details, see Set and Test System Variables in Performing Pre-installation Tasks.

Verify that the JSP files for any forms that you want to update are write-able. For example, if you want to regenerate an existing form like comment, verify that the Comments.jsp file is write-able, so that you can overwrite the existing file with a new file when you generate the new version of the form.

If you are using UNIX or Linux, verify that all *.sh files are executable.

Tomcat Errors

Tomcat does not start and displays the following error:

```
com.microsoft.sqlserver.jdbc.SQLServerException: Failed
logon:com.microsoft.sqlserver.jdbc.SQLServerException: Login failed
for user 'cascm'. The user is not associated with a trusted
SQL Server connection
```

This error message occurs if the SQL Server authentication mode is set to Windows Authentication mode and the Web Interface connection is using SQL Server authentication. To fix this problem, set the authentication mode of your SQL Server to Mixed mode, which permits both Windows Authentication and SQL Server Authentication.

Chapter 7: Installing CA Harvest SCM Reports

This section contains the following topics:

Intended Audience [BO Reports]

The following users install, upgrade, and configure CA Harvest SCM Reports:

- System administrators use the information in this guide and their operating system knowledge, to install the product for the first time, upgrade the product from release to release, and configure the product based on your implementation requirements.
- Database administrators use the information in this guide to install the database on which CA Harvest SCM Reporting runs, and configure the database for best performance.

The following persons use CA Harvest SCM Reports:

- CA Harvest SCM Administrators use the reports to measure performance. After you establish a change management process, it is important to monitor and measure the process over time to achieve the goal of continuous process improvement. Indicators that you determine help you measure processes. Over time the indicators can show as trends in reports and managers can determine if a process is becoming better or worse and take appropriate action. For example, the Projects with Activity Summary report displays a list of all projects with activity events during a selected time frame.

- Project Managers use project and package-related reports to monitor their project packages. For example, the Packages Approved report displays a list of packages that have successfully completed the Approval Process defined in the current state for the selected projects, providing basic details of each package.
- Developers use package and source-related reports to help manage their changes. For example, the Items Reserved Summary by User displays a summary of the number of reserved items for the selected user, so developers can quickly determine the items that they have checked out.

Users and User Groups for CA Harvest SCM Reports

When you install CA Harvest SCM Reports, users and user groups are defined automatically in CA Harvest SCM as follows:

User	Belongs to User Group
Harvest	CA Harvest SCM Admin
Haruser	CA Harvest SCM User

Members of the user groups *have* the following privileges with one exception:

- Create personal folders
- Create reports using a universe
- Refresh report data
- Schedule reports
- Copy, edit, and delete reports in a personal folder
- Create and edit WebIntelligence reports
- Full Control preferences
- Delete instances (Only members of the CA Harvest SCM Admin group have this privilege.)

Members of the user groups do *not* have the following privileges:

- Create folder under Public Folders
- Cannot access BusinessObjects Designer
- Create or modify a universe connection
- Delete or edit reports in Public Folders and its subfolders
- Move reports

CA Harvest SCM Reports System Requirements

CA Harvest SCM Reports has the following system requirements:

- CA Business Intelligence r3.3 for running the reports. CA Business Intelligence r3.3 is provided on a separate DVD.
- Database credentials for accessing the CA Harvest SCM database tables.

Note: For more information about CA Business Intelligence r3.3 system requirements, see the CA Business Intelligence r3.3 Release Notes at http://supportconnect.ca.com http://www.ca.com/us/support.aspx. For information about installing and configuring CA Business Intelligence r3.3, see the CA Business Intelligence Implementation Guide that is included in the bookshelf.

CA Harvest SCM Reports on Linux and UNIX

You can now deploy CA Harvest SCM Reports to a BusinessObjects server running on UNIX and Linux platforms that BOXI supports. CA Harvest SCM Reports installer remotely installs CA Harvest SCM Reports to a BusinessObjects server running on UNIX or Linux platforms. Hence, you must launch the CA Harvest SCM Reports installer from a Windows computer.

Note: You must have the Oracle client configured on the Windows computer to connect to the SCM database.

How to Install CA Harvest SCM Reports

You need BusinessObjects installed and configured before you install CA Harvest SCM Reports. You can use CA Harvest SCM Reports with CA Business Intelligence, or BusinessObjects that is not related to CA. If you already have CA Business Intelligence installed on your computer, you can directly run the CA Harvest SCM Reports installer to install CA Harvest SCM Reports.

Follow these steps:

1. Acquire a working knowledge of CA Harvest SCM and be familiar with BusinessObjects XI features as follows:
   - Read the *CA Harvest SCM Release Notes* to help ensure that your system meets the requirements for CA Harvest SCM Reports.
   - Read the *CA Business Intelligence Implementation Guide* that is included on the CA Business Intelligence installation media for information about BusinessObjects' prerequisites before installing CA Business Intelligence.

Note: For information about creating your own custom reports, see the BusinessObjects documentation.

2. Prepare your infrastructure and understand how to set up your environment. Your preparation should include the following:
   - Verify that your system meets the requirements for CA Harvest SCM Reports.

     Note: Refer to the system requirements (see page 179) section.
   - Set up server communication.
   - Select a server location.
3. Install Business Intelligence: Typical or Custom installations.

   Note: For information about installing Business Intelligence, see the *CA Business Intelligence Implementation Guide*.
4. (Custom installations only) Configure the database connection.
5. Install CA Harvest SCM Reports.
6. Deploy the reports.

   Note: For information about deploying the reports, see the *CA Business Intelligence Implementation Guide*.
7. Review the BusinessObjects documentation—BusinessObjects Enterprise InfoView User's Guide. (You can access this guide after you log in to the reports by clicking the Help icon on the main menu.)
8. Set access rights for the root folder.

More information:

BusinessObjects Services Not Running (see page 403)

Install CA Harvest SCM Reports

Perform the following procedure to install CA Harvest SCM Reports.

Follow these steps:

1. Run postinst.exe from the *DVD drive*:\Reports directory of your product distribution. Click Next.

   Note: Installer requires Java Virtual Machine. Verify that you have a compatible Java Runtime Environment installed on your computer.

   The CA License Agreement appears.
2. Read through and accept the License Agreement. Click Next.

3. Enter the BusinessObjects username, password, host name, and port number for performing the import. Click Next.

   The SCM Database Configuration page appears.

4. Select the Database type on which CA Harvest SCM is running. Click Next.

5. Click Install to install CA Harvest SCM Reports.

   The installation process starts.

   When the installation is complete, CA Harvest SCM Reports and necessary components required for running CA Harvest SCM Reports are installed.

Set Access Rights to the Root Folder

By default, the BusinessObjects administrator has access rights to the reports. The BusinessObjects denial-based access level denies access to the root folder by members of the CA Harvest SCM Admin and CA Harvest SCM User groups. This access level makes the folders and reports invisible to all members CA Harvest SCM Admin and CA Harvest SCM User groups.

Follow these steps:

1. Launch the Central Management Console using the following URL:

   `http://hostname:port/CmcApp`

2. Log in as an administrator. If the System field is blank, enter the hostname followed by CMS port number using the following format:

   hostname:*port*

   Note: The default CMS port number is 6400.

   The home page appears.

3. Click Folders, right-click All Folders, and select Properties.

   The All Folders properties appear.

4. Click User Security.

   The User Security: Root Folder page appears.

5. Click Add Principals.

   The Add Principals page appears.

6. Select CA Harvest SCM Admin and CA Harvest SCM User groups and click the right-arrow mark (> symbol). Then, click Add and Assign Security.

   The Assign Security page appears.

7. Select Folders access and click the right-arrow mark (> symbol). Click OK.

   You are able to access the reports in InfoView by logging in as harvest or haruser. You can grant access rights to the reports to any user, and make the user a member of CA Harvest SCM Admin or CA Harvest SCM User depending on privileges or role of the user.

Configure the CA Harvest SCM Database on Oracle

BusinessObjects must access the CA Harvest SCM database to run reports. You must supply the credentials for accessing the CA Harvest SCM Database. In the CA Harvest SCM Database Configuration screen, select the type of database you are using for CA Harvest SCM Release 12.5.

Follow these steps:

1. Select the Database type as Oracle.
2. In the user name column, enter the Oracle user name created for CA Harvest SCM during the database setup process (HDBSetup). By default, the user name is cascm, if you used a different user name during hdbsetup, enter it.
3. Enter the password for the given Oracle user name in the password column.
4. Enter the Oracle TNS service name in the source column.
5. Click Next to continue and finish the configuration.

   The CA Harvest SCM database is configured on Oracle.

Configure the CA Harvest SCM Database on SQL Server

BusinessObjects should be able to access the CA Harvest SCM database to run reports on it. You have to supply the credentials for accessing CA Harvest SCM Database. In the CA Harvest SCM Database Configuration screen, select the type of database you are using for CA Harvest SCM.

Follow these steps:

1. Select MS SQL Server as the Database type.
2. In the user name column, enter the SQL Server user name created for CA Harvest SCM during the database setup process (hdbsetup). By default, the user name is cascm, if you used a different user name during hdbsetup, enter it.

3. In the password column, enter the password for the SQL Server user name.
4. In the source column, enter the ODBC data source name (DSN). The ODBC data source should be configured to access the SQL Server service on which the CA Harvest SCM database is running.

   Note: The ODBC DSN should point to the CA Harvest SCM database as the default database.
5. In the server column, enter the SQL Server service name. By default, this name is the computer name on which the SQL Server is running.
6. Click Next to continue and finish the configuration.

   The CA Harvest SCM database is configured on SQL Server.

How to Set Up CA Harvest SCM Reports With Other BusinessObjects Installations

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

CA Harvest SCM Reports works with CA Business Intelligence. If you want to install CA Harvest SCM Reports with other BusinessObjects installations, do the following:

1. Use the Import Wizard to import the BIAR file for the reports to the BusinessObjects installation.
2. Configure the ODBC connection.

Import a BIAR File

If you want to use CA Harvest SCM Reports on your site's existing BusinessObjects installation, import the CA Harvest SCM Reports BIAR file to that BusinessObjects installation.

Follow these steps:

1. Insert the DVD media.
2. Select All Programs, BusinessObjects XI, BusinessObjects Enterprise, Import Wizard.

   The Import Wizard utility appears.
3. Click Next.

   The Source environment screen appears.

4. Select the Business Intelligence Archive Resource (BIAR) option from the Source drop-down list.
5. Browse to the *media drive*/Reports directory and select CA_SCM_R12_EN.biar.

   Note: In Windows, *media drive* represents the drive letter. In Linux/UNIX, *media drive* represents the mounted directory.
6. Click Next.

   The destination environment screen appears.
7. Enter the administrator password in the password field.
8. Click Next until the User and groups screen appears. Click Select All, and click Next.
9. Click Next until the Folder and objects screen appears. Click Select All, and click Next.
10. Click Next until the Finish button appears. Click Finish.

    The BIAR file is imported to BusinessObjects Enterprise.
11. Click Done.

    The wizard closes.
12. Save the universe.
13. Select File, Export.
14. Export the universe back to the same location from where it was imported.
15. Log out all the active sessions and login.

Note: Whenever the Universe parameters are modified, we recommend you to restart the CMS server and connection server to get the refreshed universe connections.

Configure the ODBC Connection

Use the Import Wizard to configure the ODBC connection so you can view the CA Harvest SCM reports that you imported to your site's installation.

Follow these steps:

1. Select All Programs, BusinessObject XI, BusinessObjects Enterprise, Designer.

   The log on screen appears.
2. Enter the Administrator password, and click OK.

3. Select File, Import.

   The Import Universe Screen appears and lists available universes.

4. Select the CA_SCM_R12_EN universe, and click OK.

5. Select File, Parameters.

   The Universe Parameters dialog appears.

6. Click Edit.

7. Enter the database user name and password for the CA Harvest SCM database.

8. Select the DSN which is configured to access the CA Harvest SCM database. If no DSN for CA Harvest SCM exists, create an ODBC DSN. Click Next.

9. Click Test Connection to test the connection for the database.

10. Click Next until the Finish button appears. Click Finish.

    The Universe Parameters dialog appears.

11. Click OK.

12. Save the universe and exit the designer.

    When you open InfoView, you are able to view and run the reports.

Chapter 8: Installing or Upgrading the Plug-In for Eclipse

This section contains the following topics:

How to Upgrade the Eclipse Plug-In From Release 12.x

If you have an existing Eclipse plug-in of Release 12.x (or its Service Packs) on your computer, perform the following tasks before installing the Release 12.5 Plug-In for Eclipse:

1. Check in or commit your Eclipse project changes to the repository before your server is upgraded to Release 12.5.
2. Recreate your projects using the Add to Workspace wizard after the Release 12.5 plug-in is installed. If you want to continue using your existing Eclipse workspace, then delete all Eclipse workspace projects that were previously shared with Harvest Release 12.x (or its Service Packs). Alternatively, you can start Eclipse with an entirely new workspace location.
3. Install the Release 12.5 client (Workbench or command line) on each computer where the plug-in will be installed.
4. Verify that you have a supported version of the Java Runtime (JRE) installed on the client.
5. Verify that you have a supported version of Eclipse installed.

   Note: If you want to upgrade to a newer version of Eclipse, you can download it from http://www.eclipse.org.
6. Proceed to install the CA SCM Plug-In for Eclipse (see page 189).

From the existing Eclipse plug-in (3.6 ,3.7 and 4.2 versions), un install the plug-in by following the instructions:

1. Choose the CA Harvest SCM Team provider plug-in and then proceed to uninstall.
2. Click uninstall.

How to Perform a First-Time Installation

If you are installing the Eclipse plug-in for the first-time, perform the following pre-installation tasks:

1. Install the CA Harvest SCM client (Workbench or command line) on each computer where the plug-in will be installed.
2. Install the Java Runtime (JRE) on your computer.
3. Install Eclipse in a location that your Release 12.5 plug-in will use specifically. Do not use this location for any prior release plug-in.

   Note: The Eclipse version must be downloaded from http://www.eclipse.org and installed on the target systems before installing the plug-in.
4. Proceed to install the CA Harvest SCM Plug-In for Eclipse (see page 189).

Eclipse Requirements

Eclipse IDE Support:

CA Harvest SCM Plug-In for Eclipse Release 12.5 supports the following releases of the Eclipse distribution.

Install any of the Eclipse Development Platforms on the supported platforms of your client computer:

- Eclipse 3.6 (Helios)

  Note: CA Harvest SCM Plug-In for Eclipse current release has been tested with the Eclipse 3.6 Classic configuration, Eclipse IDE for JEE, and Eclipse IDE for Java Developers only.
- Eclipse 3.7 (Indigo)

  Note: CA Harvest SCM Plug-In for Eclipse current release has been tested with the Eclipse 3.7 Classic configuration, Eclipse IDE for JEE, and Eclipse IDE for Java Developers only.
- Eclipse 4.2 (Juno)

  Note: CA Harvest SCM Plug-In for Eclipse current release has been tested with the Eclipse 4.2 Classic configuration, Eclipse IDE for JEE, and Eclipse IDE for Java Developers only.

The various configurations of the supported Eclipse releases are available on http://www.eclipse.org (Classic, Java EE, and so on). Download the appropriate Eclipse configuration depending on the Eclipse projects type or applications being developed.

Download and install the dependencies that are based on your Eclipse version, before installing Eclipse plug-in for Release 12.5. For Eclipse 3.6, 3.7, and 4.2 versions, download the dependencies from https://support.ca.com.

Note: During the Eclipse 3.6, 3.7, and 4.2 version dependencies installation, follow these instructions:

- For Eclipse Classic IDE and Eclipse IDE for Java Developers, select all the features from the corresponding dependency archive zip.
- For Eclipse IDE for JEE developers, select Birt and zest features from the dependency archive zip.
- For Eclipse IDE for Java Developers, download an extra .JAR (org.apache.commons.logging_1.0.4.v201101211617.jar) from the CA Support site and include it in the plugins folder of the Eclipse installation. This jar must be used to use the Dash Board Reports feature from the SCM plug-in.
- For RAD 8.0.2 and RAD 8.5, choose only the SCM plug-in (No need to install dependencies).

Install the CA Harvest SCM Plug-In for Eclipse

The CA Harvest SCM Plug-In for Eclipse is installed using the Eclipse Update Manager, an Eclipse feature. The Eclipse Update Manager is a built-in plug-in installation and update manager that simplifies the installation process and lets you manage various plug-in instances.

Note: For information about managing your configuration, see the Eclipse help.

Follow these steps:

1. In Eclipse, select Help, Install new software.

   The installation wizard appears.

2. Select the Available Software Sites tab.
3. Click Add Site.

   The Add Site dialog appears.

4. In the Location field, enter the location of the locally downloaded relevant SCMPluginDependenciesForEclipse3.x zip file.

   This points to the bundle of dependency zip files, which are prerequisites for the CA Harvest SCM Plug-In for Eclipse.

5. Click OK.

   The archived SCMPluginDependencies.zip is added to the list of available software sites.

6. Select the appropriate check boxes next to the SCMPluginDependencies.zip and click next to install the dependencies.

   Note: For more information about the check box selection, see the Dependencies selection for appropriate IDE's.

7. Click Finish, when prompted to confirm the installation.

8. Click Yes, when prompted to restart Eclipse.

   The SCM dependencies for the CA Harvest SCM Plug-In for Eclipse is installed.

9. Repeat step 4 again. In the Location field, enter the location of the locally downloaded CA Harvest SCM plug-in for r12.5.zip.

   This points to the CA Harvest SCM Plug-In for Eclipse.

10. Click OK.

    The CA Harvest SCM plug-in for r12.5.zip is added to the list of available software sites.

11. Select the check box next to the Release 12.5 plug-in version, and click next to proceed and accept the license agreement to complete the plug-in installation.

12. Click Finish, when prompted to confirm the installation.

13. Click Yes, when prompted to restart Eclipse.

    The CA Harvest SCM Plug-In for Eclipse is installed.

Chapter 9: Upgrading on Windows

Important! Upgrading these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

The Upgrade Wizard

Use the Upgrade Wizard to upgrade the CA Harvest SCM server, client, and agent from a previous product release (Release 12.0 and Release 12.1) to the current release. The Upgrade Wizard automatically detects all existing product components from supported releases on a local computer and upgrades them to the current release, in their *existing* locations. In addition, the Upgrade Wizard backs up your existing supported product release settings, forms, and configuration files and restores them for use with the current release.

The Upgrade Wizard should *not* be used to:

- Upgrade any product component from a release other than Release 12.x.
- Upgrade a CA Harvest SCM installation that uses an Ingres database.

  Note: Reconfigure an Ingres database to a supported database and then use the Upgrade Wizard to upgrade a CA Harvest SCM installation that uses an Ingres database.
- Change your database.
- Change the location of any component already installed, except for a network client.

- Change the options for any component being upgraded, or add new options. All existing options are updated and you cannot delete them.
- Upgrade the plug-in for Eclipse option.
- Specify the specific components to be upgraded.

If any of these situations apply, you must manually upgrade the component.

Note: On Windows, the product includes the SDK (HSDK), the Java HSDK (JHSDK), and the Component Object Model SDK (COM SDK). If you run the Upgrade Wizard on a computer that has the Release 12.x client installed, the Upgrade Wizard automatically installs the current release of the HSDK, JHSDK, and COM SDK while upgrading the client to the current release. This action applies to local and network clients.

More information:

Upgrade Locally Installed Components Using the Upgrade Wizard (see page 194)
Run the Upgrade Wizard in Unattended (Silent) Mode Using a Response File (see page 195)

How to Prepare for the Upgrade Wizard

Follow these steps:

1. Determine whether you run the Upgrade Wizard from your local computer, a network location, or a shared drive, rather than from the installation media.

   Note: We recommend that you run the Upgrade Wizard directly from the installation media. However, if you decide not to use the installation media, verify that you have copied all of the installation files to the appropriate location and the files are accessible.

2. If you have not already done so, read the *Release Notes*. Do not start the Upgrade Wizard until you have read that information and understand it.

   Note: You can find the most current version of the *Release Notes* at http://ca.com/support.

3. Verify that you have Windows Administrator rights on the computers that you plan to upgrade.

   Note: For information about how to verify and upgrade your rights, see your Windows documentation.

4. Verify the accuracy of your path names.

   - Path names must be standard and must begin with a drive letter, such as the C:\Program Files\ or D:\somedir\someApp\.
   - You can use mapped drives and relative path names, such as ..\somedir\someApp.
   - You cannot use Universal Naming Convention (UNC) path names, such as \\myMachine\somedir\someApp.

5. For all supported *servers* you upgrade, complete the following steps:

   - Verify that the server or client of your database is on the same computer as the Database Configuration Utility. When you use the Upgrade Wizard to upgrade the server, this utility is automatically upgraded.
   - If applicable, verify your database administrator (DBA) credentials. If your CA Harvest SCM database is running locally, when you upgrade the server, enter the user name and password of the DBA (that is, the user who has administrative rights to the database tables). Therefore, have this information ready when you run the Upgrade Wizard.
   - Back up your database.

     Note: For information about how to back up your database, see your vendor documentation.
   - Back up any custom files (including UDP scripts and forms) from the %HARVESTHOME% and %HARVESTHOME%\Forms directories.

     Note: For information about how to back up custom files, see your operating system documentation.

6. For all *local clients* you upgrade, consider the following information:

   - If you already successfully ran the Upgrade Wizard on a computer, all product components are upgraded and you do not have to run the wizard again.
   - If you installed any options such as Windows Shell Extension, when you installed any product component, the Upgrade Wizard automatically upgrades both the component and the options to the current product release at the same time.

     Important! You cannot change existing options for a component. If you want to upgrade a component, but you also want to add an option from the current release to the component, or delete an existing option from the component, you *cannot* use the Upgrade Wizard. Uninstall the component that includes the option you do not want, and then install the component again as if you were installing it for the first time.

- If you run the Upgrade Wizard on a computer that has the product's client installed, the wizard automatically installs the HSDK, JHSDK, and COM SDK while upgrading the client to the current product release.
- The plug-in for Eclipse is installed and maintained separately from the product components. Therefore, if the Upgrade Wizard detects an existing version of the plug-in, the wizard upgrades all other existing product components and options, *except* for the plug-in for Eclipse.

7. For all *agents* you upgrade, consider the following information:
   - All considerations for local clients also apply to agents.
   - You cannot use the Upgrade Wizard to upgrade a network agent. Manually upgrade the agent.
8. If you use a *response file* to upgrade the product, consider the following information:
   - All considerations for individual product components, including the server, client (including a network client), and agent, also apply to a response file.
9. Remove any emergency fixes (efixes) that have been installed on the existing release.

More information:

Upgrade Locally Installed Components Using the Upgrade Wizard (see page 194)
Run the Upgrade Wizard in Unattended (Silent) Mode Using a Response File (see page 195)

Upgrade Locally Installed Components Using the Upgrade Wizard

You can use the Upgrade Wizard to upgrade all locally installed product components, including the server, client, and agent.

Follow these steps:

1. Make sure that you have properly prepared for the upgrade.
2. Insert the installation media into your drive. If autorun is enabled on your computer, the Upgrade Wizard starts automatically.

   Note: If autorun is not enabled or if you are not running the upgrade from the installation media, start the wizard in Windows Explorer by double-clicking upgrade.exe in the folder where the wizard files have been copied.

   The license agreement appears.
3. Read and accept the terms of the agreement. Otherwise, you cannot continue.

4. Review the list of product components already installed.

   Important! If your computer has any product components installed that you do *not* want to upgrade, exit the wizard now. Then, uninstall the components and install the component again as if you were installing it for the first time.

5. Click Upgrade to start the upgrade.
6. If prompted, select the location where your computer can access the product installation files for the component being upgraded. Browse to the location and click OK.

   Note: Typically, if the wizard prompts you for the location of the installation files for the product components, you are prompted in the following order: agent, client, and server.

7. Continue entering the requested locations for all product components.
8. Continue following the on-screen instructions to complete the upgrade.

More information:

How to Prepare for the Upgrade Wizard (see page 192)

Run the Upgrade Wizard in Unattended (Silent) Mode Using a Response File

To save time during an upgrade, you can use the default response file to run the Upgrade Wizard and perform an unattended, or silent, upgrade of the existing product components on a computer.

Follow these steps:

1. Copy the default Upgrade Wizard response file (DHUR021705.rkr) to the location from where the upgrade will be run.

   Note: The DHUR021705.rkr file is provided on the installation media in the \upgrade folder.

2. Use a text editor to open the DHUR021705.rkr file and read the terms of the license agreement.
3. Accept the terms of the license agreement, as follows:

   a. Find the license agreement parameter listed in the text of the agreement.

   b. Enter the license agreement parameter as an executable option in the file and save the file.

   Important! If you do not accept the license agreement terms, the upgrade will not work.

4. Enter the -silent option either as a parameter in the DHUR021705.rkr file or enter it on the command line.

5. (Optional) Modify the response file parameters (see page 196) in the DHUR021705.rkr file based on your specific requirements.

   Note: If any option is needed for the upgrade but is not supplied in the response file, the user running the upgrade is prompted to enter the required values. For example, if the upgrade detects that the client is installed and needs to be upgraded, the upgrade checks for the corresponding entry (-clinstloc=*client-install-path-name*) in the response file. If this entry does not exist in the response file, the upgrade uninstalls the previous client and then prompts the user running the upgrade for the location of the new client installation files.

6. From the command line, enter the following command:
   `upgrade -response="pathname\DHUR021705.rkr" -silent`

   pathname

   Specifies the complete path name or relative path name to the response file DHUR021705.rkr.

   Important! If any part of the *path name* in this command includes a space, enclose the complete path name in quotation marks. For example, enclose the path name in -response=”c:\my files\temp\hupgrade.exe” in quotation marks.

   The following sample command illustrates a complete path name.

   `"C:\Program Files\CA\SCM"`

   The following sample command illustrates a relative path name.

   `"..\CA\SCM"`

   -silent

   Specifies that the response file executes silently, with no user input or notification.

7. Use the hdbsetup Database Configuration Utility to upgrade the database.

More information:

Response File Parameters (Server, Client, Agent, and Visual Studio Plug-in) (see page 196)

Response File Parameters (Server, Client, Agent, and Visual Studio Plug-in)

The following parameters for the Server, Client, Agent, and Visual Studio Plug-in apply only to the content of the DHUR021705.rkr response file that is used to run the Upgrade Wizard in unattended (silent) mode. You *cannot* use these options directly from the command line. The response file can include any combination of these parameters.

Important! When editing the response file, verify that you do not enclose parameters in quotation marks, even if they include spaces.

Example: Correct Format for Response File Parameters

This example illustrates the correct parameter for a response file.

```
-svinstloc=server-install-pathname
```

Example: Incorrect Format for Response File Parameters

This example illustrates an incorrect parameter format for a response file.

```
-svinstloc="server-install-pathname"
```

The following response file parameters are available for the Server, Client, Agent, and Visual Studio Plug-in.

-svinstloc=server-install-path-name

Specifies the path name of the installation files for the server.

-clinstloc=client-install-path-name

Specifies the path name of the installation files for the client.

-aginstloc=agent-install-path-name

Specifies the path name of the installation files for the agent.

-scmagentport=<*portnumber*>

Specifies the port number for the agent components.

-vciinstloc=vsip-install-path-name

Specifies the path name of the installation files for Visual Studio Plug-in.

Database Configuration and Maintenance

After using the Upgrade Wizard to upgrade the CA Harvest SCM server, database, and other components, configure and maintain the database. This configuration ensures that the upgraded software runs as expected in your environment. To configure and maintain the database, run the Database Configuration Utility for any supported database.

How to Prepare for a Manual Agent Upgrade

Follow these steps:

1. If you have not already done so, read the Release Notes. Do not upgrade the agent until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Write down the current version of the agent.
3. Uninstall the previous version of the agent.

More information:

Uninstall the Agent (see page 374)

Upgrade the Agent Manually

You upgrade the CA Harvest SCM agent so you can use the new product features.

Follow these steps:

1. Make sure you have completed all steps to prepare for the agent upgrade.
2. Install the agent as if you were installing it for the first time.

   The agent is upgraded.

More information:

How to Prepare for a Manual Agent Upgrade (see page 198)
Install the Agent (see page 60)

How to Prepare for the Client Upgrade Manually

Follow these steps:

1. If you have not already done so, read the Release Notes. Do not install the client until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Write down the current version of the client.
3. Back up all custom files (including UDP scripts and forms) from the %HARVESTHOME% and %HARVESTHOME%\forms directories. Make note of your backup location.
4. Back up any HClient.arg file that you use to connect to a specific product broker.
5. Uninstall the previous version of the client.

More information:

Uninstall the Client (see page 374)

Upgrade the Local Client

You upgrade the CA Harvest SCM client so you can use the new product features.

Follow these steps:

1. Verify that you have completed all steps to prepare for the client upgrade.
2. Install the client as if you were installing it for the first-time.
3. After the installation is complete, if you backed up custom files including UDP scripts and forms from your previous installation, copy these files back to the %CA_SCM_HOME% and %CA_SCM_HOME%\forms directories.

   The local client is upgraded.

More information:

How to Prepare for the Client Upgrade Manually (see page 199)
Install the Local Client (see page 51)

How to Prepare for the Server Upgrade Manually

Follow these steps:

1. If you have not already done so, read the *Release Notes* to help ensure that your DBMS version is supported. If your DBMS version does not meet the requirements listed in the *Release Notes*, upgrade it to a supported release before upgrading the server.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Write down the current version of the server.
3. Back up your database and all custom files (including UDP scripts and forms) from the %HARVESTHOME% and %HARVESTHOME%\forms directories. Make note of your backup location.
4. Uninstall the previous version of the server.

More information:

Uninstall the Server (see page 373)

Upgrade the Server

You upgrade the CA Harvest SCM server so you can use the new product features.

Follow these steps:

1. Make sure you have completed all steps to prepare for the server upgrade.
2. Install the server as if you were installing it for the first time.
3. After the installation is complete, if you backed up custom files including UDP scripts and forms from your previous installation, copy these files back to the %CA_SCM_HOME% and %CA_SCM_HOME%\forms directories.

   The server is upgraded.

More information:

How to Prepare for the Server Upgrade Manually (see page 200)
Install the Server (Typical Installation) (see page 30)
Install the Server (Custom Installation) (see page 32)

Chapter 10: Upgrading on UNIX, Linux, and zLinux

Important! Upgrading these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

How to Prepare for the Server Upgrade

An *upgrade* means that you install the CA Harvest SCM Release 12.5 server software, and then you upgrade the database.

If you specify CA_SCM_HOME to the existing install directory for the installation of the CA Harvest SCM Release 12.5 server, the following actions automatically occur:

- arg files are updated.
- existing dfo files are converted to eTPKI format.

If you install CA Harvest SCM Release 12.5 to a separate directory, you must run svrenc to encrypt the passwords for the server, LDAP, and user-defined dfo files in the CA_SCM_HOME directory.

Follow these steps:

1. Back up your database.

   Note: For information about how to back up your database, see your vendor documentation.

2. Back up any custom files (including custom UDP scripts and custom forms) from the $HARVESTHOME and $HARVESTHOME \forms directories if you are installing to the same location as the previous installation.

   Note: For information about how to back up custom files, see your operating system documentation.

3. Install or upgrade the Public Key Infrastructure (eTPKI).
4. Install or upgrade PEC.
5. Install or upgrade ODBC.

More information:

Install CAPKI for All the Users on a Computer (see page 85)
Install Enterprise Communicator (PEC) (see page 91)
Install CAI/PT ODBC (see page 89)
Manual Upgrade of the Database (Oracle) (see page 221)

Extract the Installation Files

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Complete these steps before performing an upgrade.

Important! Do the following before you start these steps:

- On Linux, verify that the umask is set to 0022.
- On UNIX, verify that the umask is set to 022.
- Log on as the *cascm* user (the user named *cascm*).

1. Insert the installation media for your UNIX or Linux operating environment into the drive. Mount the drive if necessary. The following instructions use /cdrom as the mount point.
2. Change to the installation directory. For example, if you are using the default installation directory, enter the following command:

   ```
   cd /opt/CA/scm
   ```

3. Enter the following command to copy the installation files for your UNIX or Linux operating environment to the current directory. Verify that you include the space and period at the end of the command; they represent the current directory.

   ```
   cp /cdrom/bin/directory/scm.tar.gz .
   ```

   directory

   Specifies the directory for your UNIX or Linux operating environment.

4. (AIX operating environments only) As the root user, run slibclean to clear all cached lib executables so that the lib files are available for overwrite during the extraction process.

5. Enter the following commands to extract the files:

   ```
   gunzip scm.tar.gz
   tar xvf scm.tar
   ```

   The installation files are extracted.

Upgrade a Server (Local and Remote Oracle Database)

You can *upgrade* a CA Harvest SCM release 12.x server running a local or remote Oracle database to the current release of the product. After extracting and installing the tar files, perform the following steps to upgrade the server.

Important! Before you begin, verify that the product broker, server, and agent processes are not running.

Follow these steps:

1. After extracting the Release 12.5 tar files, navigate to the product installation directory. For example, enter the following command:

   ```
   cd /opt/CA/scm/install
   ```

2. Enter the following command to run the installation script:

   ```
   ./install.sh
   ```

3. From the installation options, select option [2] Upgrade SCM. This option upgrades the product from release 12.x to the current release.

4. Specify whether to install the CA Software Delivery Integration.

   Note: CA Software Delivery is required to *use* this feature but is not required to *install* this feature.

   - If No, enter N and skip to the next step.
   - If Yes, enter Y and complete the related fields. For details about these fields, see CA Software Delivery Integration Parameters (see page 97). When the confirmation page appears, verify the values specified and continue.

5. Specify the method the product server and agent will use to authenticate users' logon credentials.

   Internal

   Uses internal (CA Harvest SCM) authentication. Login credentials provided to the broker are validated against the internal product user data.

   If you select Internal, skip the rest of this step and continue at the next step.

   OpenLDAP

   Uses an external server. Login credentials provided to the broker are validated against the external authentication server. If you select OpenLDAP authentication, you are prompted to supply the required LDAP-related information. These required LDAP values and optional values appear in confirmation pages so that you can verify them. Optional fields are automatically filled with default values. You can change them in the confirmation pages as needed.

   For details about these fields, see LDAP Compliant Directory Configuration Parameters (see page 101).

   Note: When you install LDAP authentication, the OpenLDAP and OpenSSL open source libraries are installed automatically in the product folders, if they are not already installed. For information about OpenLDAP, see the OpenLDAP web site. For information about OpenSSL, see the OpenSSL web site.

   Mixed Mode authentication

   Lets the SCMAdmin create users internally even though the authentication mode may be set to External (LDAP).

   Note: Mixed Mode authentication does not use LDAPserver for Authentication if users are created internally.

   External User Group authentication

   Lets the SCMAdmin define LDAP support for external user groups.

6. Specify that you are using an Oracle database.
7. The values for CA_SCM_HOME and DBMS-related environment variables appear. When prompted, select the appropriate number to modify a value; otherwise, accept the default [0] to continue.
8. If the eTrust Public Key Infrastructure (CAPKI) location is not available you may be prompted to enter the path to the CAPKI installation directory. If prompted, enter the complete path to the eTPKI installation directory. If not prompted, go to the next step.
9. If RTHOME is not set, you are prompted to enter the Enterprise Communicator (PEC) installation directory. If you are not prompted, the installation uses the current value of the environment variable, $RTHOME. If prompted, enter the Enterprise Communicator installation directory; the default is /opt/CA/pec.

10. To run the product server from behind a firewall, you must specify a range of available ports, as follows:

    At the Firewall port range prompt, specify the port range by entering the starting port number, a comma, and the ending port number. (Spaces are optional and are ignored.) For example, to specify a port range of 1500 through 1502, enter the following at the prompt:

    ```
    1500, 1502
    ```

    Important! The number of ports in the range must be greater than or equal to the maximum number of server processes and remote agents running behind the firewall.

11. The Database Configuration Utility (hdbsetup) starts. Use this utility to configure your database. This utility helps you to upgrade the database schema but does not remove or overwrite existing product data. You can optionally run the utility now or later, but you must run the utility before you can start using the product.

    Important! If you are using Oracle, do not use configdsn to set up your CA Harvest SCM database. You must use the Database Configuration Utility instead.

    Check the $CA_SCM_HOME/install/log directory for log files created during the upgrade.

    Note: After the installation is complete, if you backed up custom files including UDP scripts and forms from your previous installation, copy these files back to the %CA_SCM_HOME% and %CA_SCM_HOME%\forms directories.

More information:

LDAP Compliant Directory Configuration Parameters (see page 101)
Extract the Installation Files (see page 93)
The hdbsetup Database Configuration Utility (see page 223)

Upgrade the Command Line Utilities

You upgrade the CA Harvest SCM command-line utilities so you can use the new product features.

To upgrade the command-line utilities, do *one* of the following:

- Make the files in $HARVESTHOME writable and install the new version of the command line utilitiescommand-line utilitiesas the existing version.

  Note: You need to regenerate the .dfo files if any exist for using the svrenc utility.
- Delete the existing version of the command-line utilities and then install the new version in the new location.

The command-line utilities are upgraded.

Note: Back up all custom files from %HARVESTHOME% and %HARVESTHOME%\forms directories. Make note of your backup location.

More information:

How to Prepare for the Client Components Installation (see page 120)

Upgrade the Agent

You upgrade the CA Harvest SCM agent so you can use the new product features.

Important! If you are upgrading from a previous release and you are running the agents on *remote* computers, you must install the agents on the remote computers. If the agents are already installed, follow this step.

To upgrade the agent, do *one* of the following:

- Make the files in $CA_SCM_HOME writable and install the new version of the agent in the same location as the existing version.

  Note: You need to regenerate the .dfo files if any exist for using the svrenc utility.
- Delete the existing version of the agent and then install the new version in the new location.

The agent is upgraded.

More information:

Install the Agent (see page 130)

Chapter 11: Upgrading on z/OS

Important! Upgrading these CA Harvest SCM components is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

Upgrade the Agent

You upgrade the CA Harvest SCM agent so you can use the new product features.

Note: When you upgrade from Release 12.x to the current release, the USS agent is overwritten with the z/OS agent. The z/OS agent is a single executable that runs as a daemon process in USS. The z/OS agent supports both HFS and MVS partitioned data sets (PDS).

Follow these steps:

1. Delete any existing MVS agent installations.
2. Install the agent as if you were installing it for the first time.

More information:

How to Install the z/OS Agent (see page 139)

Chapter 12: Upgrading Form Types

Important! Upgrading form types is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

Form Type Customization Upgrade

In a previous release of CA Harvest SCM (Release 12.x), you may have built upon the default form types supplied with the product or created a *customized* version of a form type to suit your needs. When you upgrade and install the most current release of the product, you want to help ensure that all of your form type customization is preserved and successfully saved in the product database table on the product server, so your users can use the customized forms.

In the current release of the product, JavaScript, rather than Visual Basic, is used as the scripting language for default values, initialization, validation, and interaction of form fields. Therefore, you must be proficient in JavaScript to successfully convert your existing customized form types using Visual Basic scripts for use in the current release. In addition, XML is used to store the form definition (HFD) file in the product database table on the product server. Therefore, it is also helpful to be proficient in XML.

How to Convert Customized Form Types and Add Them to the Database

To successfully convert your customized form types from a previous release of CA Harvest SCM (Release 12.x) to the current release of the product, so Web Interface and Workbench users can use the customized form types, you must do the following to help ensure your success:

1. Verify that all Form Definition (HFD) files that you want to convert are located in the form reference directory on the server.

2. Convert the HFD files to XML format using the Custom Form Converter.

   Note: For information about creating and modifying form types, see the *Administrator Guide*.

3. Modify the form type XML files in the form reference directory on the server to use JavaScript. This step is required if you used Visual Basic scripting for default values, initialization, validation, and interaction of form fields.

   Note: For information about modifying a form type by editing the XML file, see the *Administrator Guide*.

4. Add the customized form types to the database using the hformsync command (see page 212).

   Note: Repeat steps 3 and 4, as necessary, to fine-tune your forms and add them to the database.

5. (Workbench users) Add the form types to the database using the hformsync command.

   Note: For information about adding form types to the database for use in the Web Interface, see the *Administrator Guide*.

6. (Web Interface users) Add the form types to the database using the Generate Form Page of the Web Interface.

   Note: For information about adding form types to the database for use in the Workbench, see the *Administrator Guide*.

Add the Customized Form Types to the Database

After you convert the form definition files (HFD) files in the form reference directory to XML format and edit the XML files to use JavaScript, you must add the customized form types to the database. The hformsync command-line utility lets you add the customized form types to the database.

Follow these steps:

1. At the command prompt, navigate to the %AllUsersProfile%\Application Data\CA\SCM\Forms directory.

2. Run the hformsync command, select the *-f* option, and any additional option. For example:

   `hformsync -b brokername -usr harvest -pw harvest -d path of the forms folder -f xml file`

   Note: For more information about the hformsync command, including a description of all command options, see the *Command Line Reference Guide*.

   All customized form types in XML format are added to the CA Harvest SCM database and will display in the Workbench and Web Interface.

Chapter 13: Upgrading the Web Interface

Important! Upgrading the Web Interface is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

This section contains the following topics:

How to Prepare for the Web Interface Upgrade

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Follow these steps:

1. If you have not already done so, read the Release Notes and Readme (if one is provided). Do not upgrade the Web Interface until you have read that information and understand it.

   Note: You can find the *Release Notes* at http://ca.com/support.

2. Determine the Web application server type.

   Note: For information about support application server types, see the Release Notes.

3. Determine the full path to the existing Web Interface installation (that is, Harweb home).

   Note: For WebSphere, this is the directory where the harweb.war file is deployed.

   For example, Tomcat on Windows:

   ```
   C:\Program Files\Apache Software Foundation\Tomcat 6\webapps\harweb
   ```

   For example, Websphere on Windows:

   ```
   C:\Program
   Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\computer-nameNo
   de01Cell\harweb.ear\harweb.war
   ```

   For example, Tomcat on Linux/UNIX:

   ```
   .../Tomcat/apache-tomcat-6.0.2
   ```

For example, Websphere on Linux/UNIX:

```
.../IBM/WebSphere/AppServer/profiles/AppSrv01/installedApps/computer-nameNode01Cell/harweb.ear/harweb.war
```

4. Determine the temporary directory to be used by the installation (a default is provided by the wizard).
5. Determine the deployment directory (that is, the full path to the web application server's home directory).

Upgrade the Web Interface (Windows, UNIX, and Linux)

You upgrade the Web Interface so that the new product features can be used. Users who need to upgrade their Web Application servers to meet the system requirements of this CA Harvest SCM release cannot upgrade. Instead, they must uninstall the existing Web Interface and then do a new installation.

Important! Before upgrading the Web Interface, verify that you have access rights to create directories and files in this location.

Follow these steps:

1. Start or stop your application server.
   - (WebSphere) This application server must be running during the upgrade.
   - (Apache Tomcat and JBoss) These application servers must be shut down during the upgrade.
2. (Optional) If you have customized any CA Harvest SCM JSP pages in your existing Web Interface installation, back them up to a different location.

   Note: Do not copy the previously customized JSP pages back into your upgraded product installation. You should *only* use these JSP pages as a reference to customize the upgraded versions.
3. Uninstall the previous version of the product client.
4. Verify that the current version product client is installed.
5. (WebSphere *only*) Open <*harweb-home*>/WEB-INF/harweb.cfg and manually add the entries for SCM Database user and password as follows and then restart the WebSphere application server:

   User=<*SCM-Database-User*>

   Password=<*SCM-Database-Password*>

6. Start the installation by inserting the installation media in your drive.
7. Click Upgrade to start the upgrade.
8. Read the *Release Notes* to help ensure that the application server you are using is supported by the product. If necessary, uninstall your existing application server and install a supported version.
9. Deploy the harweb.war file to the application server.

   Important! When you install the Web Interface, you can enter non-English characters (such as Japanese or accented characters) when specifying the name of the context root (.war file name). However, if you use non-English characters in the name of the .war file, your application server may encounter errors, and may not be able to generate a context with a matching name, because of system limitations in some servlet containers. Therefore, we strongly recommend that you only use single-byte, English characters when naming the .war file.
10. (WebSphere *only*) Open <*harweb-home*>/WEB-INF/harweb.cfg and remove the SCM Database user and password as follows, execute svrenc to create harweb.dfo file, and then restart WebSphere application server:

    - Windows:

      ```
      svrenc –f harweb.dfo –dir "<harweb-home>\WEB-INF"
      ```

    - UNIX, Linux:

      ```
      svrenc –f .harweb.dfo –dir "<harweb-home>/WEB-INF"
      ```

    The Web Interface installation is upgraded.

Chapter 14: Configuring the Database

Important! Configuring CA Harvest SCM is one step in the overall implementation process. To understand all of the steps you must follow for a successful implementation, see How to Implement the Product (see page 16).

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

What You Need to Know

Running the Database Configuration Utility requires the skill sets of a database administrator and operating system user. These responsibilities may fall to one or more individuals. If necessary, consult the appropriate personnel at your site for assistance in running the Database Configuration Utility. Before you install, upgrade, or configure the CA Harvest SCM database, perform any tasks that apply to your environment.

Using Multiple Databases

A CA Harvest SCM server connects to only one database at a time. The product stores and uses only one user name and password and passes these logon credentials to a pre-configured ODBC connection.

Therefore, if you plan to use the product with more than one database, you must set up server versions, with each product server connecting to a distinct database instance.

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

Required Software

Verify that the server or client of your DBMS is installed on the same computer as the Database Configuration Utility.

Check the Release Notes for the DBMS version supported by the product. Verify that your DBMS is installed or upgraded based on the requirements.

Note: For instructions to install or upgrade Oracle or SQL Server, see your Oracle or SQL Server documentation.

The DBMS \bin Directory

Verify that the bin directory for your DBMS is included in the PATH environment variable. The following are sample bin directory names:

- Oracle 10g: %ORACLE_HOME%\ora10\bin
- SQL Server 2005: %SQLSERVER_HOME%\90\Tools\Binn

The Database Authentication Method (SQL Server)

If your DBMS is SQL Server, decide whether to use Windows authentication or SQL Server authentication for the CA Harvest SCM database connection. Before you decide, verify the authentication method that your server uses.

If your server is set to Windows authentication, then you *must* set your product database connection to use Windows authentication. If your server is set to Windows and SQL Server authentication (mixed mode), then you can set your product database connection to use either Windows authentication or SQL Server authentication.

Note: For instructions to check and change the authentication mode, see your SQL Server documentation.

The System Administrator Role and the Windows User ID (SQL Server)

Important! For SQL Server, verify that the Windows operating system user name you use to create the CA Harvest SCM database is granted the system administrator role in SQL Server. This requirement exists because the product database installation scripts use the SQL Server osql utility running with a trusted connection. For instructions to grant this role, see your SQL Server documentation.

How to Connect to a Remote Database (Oracle)

Important! This procedure applies to Oracle only.

Note: The instances of Linux in this section refer to both the Linux and z/Linux operating environments.

If the CA Harvest SCM server and the Oracle database reside on the same computer, skip this procedure. If you are using a remote Oracle database, perform these steps if you have *not* already connected to the remote Oracle database from the product server.

This procedure describes how to connect to a remote Oracle database from the product server on *Windows*. For instructions to connect to a remote Oracle database from the product server on *UNIX or Linux*, see your Oracle documentation.

You can define a default service on any Oracle server that is supported by the product. Doing so enables co-resident ODBC applications such as the product server to connect to a default service while bypassing the SQL*NET and TCP/IP layers. The result is a significant performance improvement.

Use the steps in this procedure a*s a model* to define an Oracle service name for the product ODBC datasource. These steps use the Oracle Net Configuration Assistant that is supplied with Oracle 9i. For the exact steps that you perform, see the Oracle documentation for the version that you have installed.

1. On the computer that is selected to run the product, from the Windows Start menu, select Programs, Oracle, Configuration, and Migration Tools, Net Configuration Assistant.
2. On the Welcome dialog, select Local Net Service Name configuration. Click Next to continue.
3. The next dialog prompts you to select an action: Click Add, and click Next to continue.

4. On the next dialog, select the option that matches the version of Oracle that is installed at your site. For details, see your Oracle documentation.

   Click Next to continue.

5. The next dialog prompts you to specify a service name for your Oracle database service name. The service name is usually its global database name. Enter the service name and click Next to continue.

6. On the next dialog, select the network protocol to use for accessing the database. Choices include TCP, TCPS, IPC, and NMP. Select the protocol and click Next to continue. In this session, we select TCP.

   The next dialog prompts you for the host name and port number of the database computer.

7. Enter the host name and specify the port number; typically, specify the standard port number. However, if you are using Tomcat, note the following warning:

   Important! Do not use the same port number for the Tomcat and Oracle database servers. By default, the Tomcat server uses port 8080, and if the Oracle database also uses the same port, it throws an error.

8. The dialogs for each network protocol vary slightly; if applicable, supply any other information that is requested on this dialog. Click Next to continue.

9. The next dialog asks whether to test the connection between the product and the database. Click Yes to perform the test or No to skip the test. We strongly suggest that you click Yes.

10. If you clicked Yes in the previous step, the next dialog displays a message stating whether the connection test was successful. If the test fails, you may need to change the user name and password used to log in to the database. To do so, click Change Login and supply the requested information. If the test fails again after you change your logon credentials, click Help for further assistance.

    When the test succeeds, click Next to continue.

11. On the next dialog, leave the field blank to accept the default (the service name you specified earlier in Step 5) as the net service name. Accept the default or enter a different name. Click Next to continue.

12. On the next dialog, specify No when prompted whether to configure another net service name, because the product uses one database only. Click Next.

13. Click Finish to close the Oracle Net Configuration Assistant.

How to Create the Database Manually (After Custom Installation)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

If you are using Oracle or SQL Server as your DBMS, you must run the Database Configuration Utility after you have installed your DBMS and the CA Harvest SCM server. At the end of the product server custom installation, you can optionally start the Database Configuration Utility. To start the utility on Windows, click Launch the Database Configuration Utility. To start the utility on UNIX or Linux, answer Yes when you are prompted to run the utility to create the database. If you did not run the Database Configuration Utility after the product server installation, you can start the utility at anytime from the command line.

Follow these steps:

1. Create the product repository.
2. Configure ODBC DSN.
3. Encrypt the user name and password for accessing the product database.

Manual Upgrade of the Database (Oracle)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You upgrade the CA Harvest SCM database *manually* on Windows, UNIX, or Linux, when you upgrade an existing product server running Oracle. You must run the Database Configuration Utility manually to upgrade your product database.

If your CA Harvest SCM database is running on Oracle, see the product's Release Notes for the supported Oracle versions. Use that information to decide whether to upgrade the version of Oracle you have installed.

Important! If you need to upgrade your Oracle version, you must do so before you upgrade or install the product server. For instructions to upgrade Oracle, see your Oracle documentation.

If you are upgrading a product database running on Oracle, after you have upgraded your DBMS (if necessary) and the product server, run the Database Configuration Utility to upgrade the product database.

If your product database is running on Oracle and you do not use the Windows upgrade wizard to upgrade your product server, upgrade your database using the Upgrade SCM Repository option of the Database Configuration Utility.

More information:

How to Configure the Repository Using the hdbsetup Database Configuration Utility (see page 235)

How to Connect a Server to a Different DBMS Server

If you have already set up a different CA Harvest SCM database on either a different DBMS server or the same DBMS server, and you want to change your server to use this database, perform the following actions using the Database Configuration Utility:

1. Configure the ODBC DSN.
2. Create the database user (SQL Server only)

   Important! Perform this step only if you are using SQL Server with Windows authentication. This step creates a database user with the same name as your operating system user and grants this new database user administrative access to the database.
3. Encrypt the database user name and password.

More information:

Database Configuration and Maintenance (see page 197)

Database Deletion

You can use the Database Configuration Utility (see page 197) to delete the CA Harvest SCM database.

How to Connect to the Database with a New User (SQL Server)

The Database Configuration Utility lets you create a CA Harvest SCM database user *only* if your DBMS is SQL Server. If you are using SQL Server, consider the following information to determine whether to create a database user.

If you want to use Windows authentication to connect the server to a local or remote server and you do not want to use the same user account that installed the database, you can *optionally* create a database user to connect the product to the database.

For example, suppose you install the server and create the database as the operating system user named *administrator* on the computer named *testserver*. However, you do not want to use that user to run the server. Instead, you want to use the operating system user named *cascm* to start the server. In this example, use the Database Configuration Utility to create a database user named *testserver\cascm*. The new database user is used to enable the server to connect to SQL Server with the required access rights.

To create a database user and use it for your database connection, use the Database Configuration Utility to perform the following tasks:

1. Create a database user.
2. Encrypt the database user name and password.

More information:

How to Configure the Repository Using the hdbsetup Database Configuration Utility (see page 235)

The hdbsetup Database Configuration Utility

Use the hdbsetup Database Configuration Utility for the following reasons:

- Configure your CA Harvest SCM database
- Upgrade your product database from the 12.x release
- Load project template information
- Load XML form templates

Run the utility after your database software is installed and either *during* or *after* the CA Harvest SCM server installation. During the product server installation, you are automatically prompted to run the utility after the prerequisite tasks are completed. The utility records all its activities in the hdbsetup.log file in the CA_SCM_HOME\log directory**.**

You can run the hdbsetup Database Configuration Utility in either interactive mode or command-line mode. The utility runs from the command prompt in both modes, even during the initial CA Harvest SCM server installation. Both modes require you to supply the same information in different formats.

More information:

How to Configure the Server After Installation (see page 109)
Find the SQL Server Authentication User for Database User (see page 152)

Before You Run the hdbsetup Database Configuration Utility

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Before running the hdbsetup Database Configuration Utility, perform these steps:

1. (UNIX and Linux) Verify that the operating system user running the utility has write permission to the $ODBC_HOME/odbc.ini file.

   Note: If necessary, see your system administrator or operating system documentation for instructions.

2. (for Windows installations using Oracle Database 10g or 11g 64-bit editions only) Verify that you installed the appropriate Oracle Database 10g/11g Client for Microsoft Windows (32-bit) on the server *before* you installed the CA Harvest SCM Release 12.5 server.

3. Verify that the CA Harvest SCM server and DBMS have been installed or upgraded to the version that the product supports.

Connect to Your Local or Remote DBMS Server (Oracle)

If your DBMS is *Oracle*, CA Harvest SCM uses the service name to connect to local or remote Oracle server. When prompted by the hdbsetup Database Configuration Utility for the service name, do the following:

- For a local DBMS server, press Enter to use a blank service name.
- For a remote DBMS server, enter the service name that points to the remote server.

  Note: For instructions to create the service name, see Connect to a Remote Database (Oracle) (see page 219).

Interactive Mode

Interactive mode prompts you to enter information one parameter at a time until you are finished with a task. Use this mode for any of the following situations:

- There are no command-line arguments.
- The option -interactive occurs as the first command-line argument.

Logging Into Your Database (Interactive Mode)

When using the hdbsetup Database Configuration Utility in interactive mode, when prompted, log in to your database by entering the user name and password (logon credentials) for the database administrator. For some operations, you are prompted to enter the logon credentials for the CA Harvest SCM database user. If the product database user that you enter does not exist, then the Database Configuration Utility creates the user and grants it administrative access to the product tables. These logon credentials are used for certain database operations.

If your DBMS type is SQL Server and you use Windows authentication to connect, your logon credentials are authenticated through the operating system. Therefore, verify that your Windows operating system user ID is granted the system administrator role in SQL Server.

For enhanced security, you must supply these values every time you start the utility and attempt to execute a command from the Configuration Menu. Typically, you enter your logon credentials once per session. The utility does not store your logon credentials for use in future sessions.

The equivalent options for the DBMS administrator login for command line mode are:

`-ausr=`*`username`* `-apwd=`*`password`*

In this command, *username* and *password* are the logon credentials of the DBMS administrator.

The equivalent options for the product database user login for command line mode are:

`-husr=`*`username`* `-hpwd=`*`password`*

In this command, *username* and *password* are the logon credentials of the product user who has administrative access to the product database.

In command line mode, you must specify the database type option and your logon credentials each time you enter a command at the command prompt. If you are using a response file, you must specify the database type option and your logon credentials once in the file.

Run hdbsetup in Interactive Mode

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

When you have more than one DBMS installed, you can configure only one DBMS at a time.

Note: To exit the hdbsetup Database Configuration Utility at any time, press Ctrl+C.

Follow these steps:

1. Do *one* of the following:
   - Run the CA Harvest SCM server installation and, when prompted, select to configure your database.
   - (Windows) Run the hdbsetup.exe file in the CA_SCM_HOME directory.
   - (UNIX and Linux) Run ./hdbsetup (no file extension) in the CA_SCM_HOME/bin directory.

   The command prompt appears and the utility starts.
2. (Windows) Enter the first character of the type of DBMS you want to configure:

   ```
   [O]racle, [S]QL Server
   ```

   Note: On UNIX, the DBMS type is set to Oracle by default, because CA Harvest SCM on UNIX supports Oracle only.

   The Configuration Menu appears.
3. At the ENTER DESIRED ACTION prompt, enter the two-character option for the task you want to perform. If you enter an option incorrectly or if you do not enter all the options required, the utility prompts you to supply additional information.

Note: Most of these operations require additional information or other operations, such as logging in with a user name that has administrative access. The Database Configuration Utility lets you perform several operations at any time on your DBMS for testing, maintenance, and so forth.

Command-Line Mode or Response File

Command-line mode accepts parameters on the command line or from a response file and prompts you for missing parameters, unless you specify the -noprompt option.

The -noprompt option in command-line mode is used primarily in typical installations and response file installations. The typical installation automatically specifies the -noprompt option and supplies preset default values.

You may optionally use the -noprompt option explicitly from the command prompt or in response files. However, if you do not enter a required parameter in command-line mode, an error message appears and is recorded in the hdbsetup.log file.

hdbsetup Command Syntax Rules

When you run the hdbsetup Database Configuration Utility using command-line mode or through a response file, follow these syntax rules.

1. Use dashes when specifying the option format. For example, specify *-noprompt*, not *noprompt*.

   Some options are expressed in the format *-parameter=value*, for example:

   ```
   -husr=jjones
   ```

   Note: The options are not case sensitive. In addition, the order in which options are specified does not matter.

2. Specify only *one* value for each parameter, for example:

   ```
   -co -dsn=cascm
   ```

   The following command is intended to create the ODBC data source using two database names. However, it creates the data source name *cascm* because only one value for the -dsn parameter is allowed. The last value specified is used.

   ```
   -0 -co -dsn=mdb -dsn=cascm
   ```

3. (Oracle) When running the Database Configuration Utility from the command prompt or through a response file, you cannot use spaces when specifying the values for variables, including user names. At the command prompt, if the value for a parameter contains spaces, enclose the value in quotation marks. For example, to test the repository with CA Harvest SCM user Latesha Smith, enter the following command:

   ```
   -0 -ausr=username -apwd=password -tr -huser="Latesha Smith" -hpwd=password
   -svc=servicename
   ```

   Note: At the command prompt, quotation marks *cannot* be used as part of a parameter value. This restriction does not apply to response files.

4. (Oracle) When running the Database Configuration Utility from the command prompt or through a response file, you cannot use spaces when specifying the values for variables, including user names. When using a response file, do not enter quotation marks around parameters that contain inner spaces. Any quotation marks encountered in a response file are considered part of the submitted parameter. For example, in a response file, to create the product database user Bob "CA Harvest SCM Admin" Smith, enter the following command:

   ```
   -husr=Bob "CA Harvest SCM Admin" Smith
   ```

5. You can optionally submit multiple commands in any order at the command prompt. Regardless of the order in which you type the commands, the utility always executes them in the following order:

   ```
   CR  Create SCM Repository

   CO  Configure ODBC DSN

   TR  Test For SCM Repository Existence

   UR  Upgrade SCM Repository

   CU  Create SCM Database User

   DR  Delete SCM Repository

   CM  Convert MDB to Standalone

   EP  Encrypt DB Username and Password


   LP  Load Projects

   LF  Load Forms
   ```

   Important! As noted previously, each *parameter*=*value* expression can be submitted only *once*. If a parameter is submitted twice, the last value submitted is used.

hdbsetup Command Options (Oracle)

The same two-character options listed in the hdbsetup Configuration Menu apply to command-line mode. In command-line mode, precede each option with a dash; for example, -tr to test a repository. The following commands are available for Oracle databases:

co

Configures ODBC DSN

tr

Tests for CA Harvest SCM repository existence

dr

Deletes CA Harvest SCM repository

cr

Creates CA Harvest SCM repository

ur

Upgrades CA Harvest SCM repository

ep

Encrypts DB username and password

lp

Loads project templates

lf

Loads XML form templates

noprompt

Specifies silent mode

ausr=*username*

Specifies the database admin user name

apwd=*password*

Database admin password

husr=*username*

Specifies the CA Harvest SCM database user name

hpwd=*password*

Specifies the CA Harvest SCM database user password

svc=*service*

Specifies the Oracle service name

dvr=*driver*

Specifies the driver name

dsn=*datasourcename*

Specifies the ODBC data source name

mfile=*fullname*

Specifies the meta tablespace file full name

bfile=*fullname*

Specifies the BLOB tablespace file full name

ifile=*fullname*

Specifies the index tablespace file full name

msz=*sizeMB*

Specifies the meta initial tablespace size

bsz=*sizeMB*

Specifies the BLOB initial tablespace size

isz=*sizeMB*

Specifies the index initial tablespace size

tsn=*name*

Specifies the temp tablespace name

rbn=*name*

Specifies the rollback tablespace name

rsp=*filename*

Specifies the response file

rpwd=password

Specifies the CA Harvest SCM database password for the harrep user

Note: The harrep user is read-only user used internally by the reporting tools.

h

Displays Help

x

EXITs

hdbsetup Command Options (SQL Server)

The same two-character options listed in the hdbsetup Configuration Menu apply to command-line mode. In command-line mode, precede each option with a dash; for example, -tr to test a repository. The following commands are available for SQL Server databases:

co

Configures ODBC DSN

cu

Creates CA Harvest SCM database user

tr

Tests for CA Harvest SCM repository existence

dr

Deletes CA Harvest SCM repository

cr

Creates CA Harvest SCM repository

ur

Upgrades CA Harvest SCM repository

cm

Convert MDB to stand-alone

Important! This option deletes all the MDB tables that are not owned by CA Harvest SCM. If you are using the MDB with other products, do *not* use this option.

ep

Encrypts DB username and password

lp

Loads project templates

lf

Loads XML form templates

noprompt

Specifies silent mode

dbn=*dbname*

Specifies the database name

dsn=*datasourcename*

Specifies the ODBC data source name

ausr=*username*

Specifies the database admin user name

apwd=*password*

Specifies the database admin password

husr=*username*

Specifies the CA Harvest SCM database user name

hpwd=*password*

Specifies the CA Harvest SCM database user password

svr=*server*

Specifies the SQL Server server name

conmethod=[Windows|Sqlserver]

Specifies the SQL Server connect method

dvr=*driver*

Specifies the driver name

domain=*server*

Specifies the domain name

rsp=*filename*

Specifies the response file

h

Displays Help

x

EXITs

Run hdbsetup in Command-Line Mode

When you run the hdbsetup Database Configuration Utility in command-line mode, provide *all* parameters on the command line or in a response file. At the command prompt or in the response file, enter *all* the options for all tasks that you want to apply to your database. If you do not specify -noprompt and a required parameter is missing, you are prompted to enter it. If you specify the -noprompt option and do *not* provide all parameters for the operations you have specified, you receive an error message and the utility stops running.

To run the hdbsetup utility in command-line mode from the command prompt, enter the following command:

```
hdbsetup options
```

Note: For a complete list of all the options and arguments to use at the command line or in a response file, enter the following command:

```
hdbsetup -h
```

Example: Create the ODBC DSN (Oracle)

This example shows how to use the hdbsetup Database Configuration Utility to create the ODBC DSN for Oracle.

```
-0 -co -svc=svcname -dsn=datasourcename -dvr=drivername
```

When you use the utility, specify the -co option and provide the following information:

- The database type: -O
- The data source name: -dsn=*datasourcename*
- The Oracle service name: -svc=*servicename*
- The Oracle driver name: -dvr=*drivername*

Run hdbsetup from a Response File

To run the hdbsetup Database Configuration Utility in command-line mode from a response file, use the following command:

```
hdbsetup -rsp=filename
```

In this command, *filename* is the name of your response file. If the file exists in the current working directory, specify only the file name (without the path name). If the file exists in a different directory, specify the complete path name. Enclose the value in quotation marks if it contains spaces.

Each line in the response file must include one parameter, or one parameter=*value* setting. In addition, you must specify a parameter *only once* in a response file. If the same parameter is specified more than once, then the last parameter specified is used.

Example: Set the Temporary Tablespace Name Twice

In this example, if you include the following lines in the response file to set the temporary tablespace name twice, only the last setting specified (-tsn=TEMP1) is used.

```
...
 -tsn=TEMP
...
...
...
...
 -tsn=TEMP1
```

Important! Before writing and using a response file make sure that you understand the functions performed by each operand used in response files.

Example: Configure the ODBC DSN (Oracle)

In this example, the sample response file configures the ODBC DSN (-co) for Oracle (-o).

```
-o
-co
-svc=cascmsvc
-dsn=cascm
-dvr=Oracle in OraDb10g_home1
```

Example: Create the Repository, Tablespaces, and Database User Name and Password (Oracle)

In this example, the sample response file creates the CA Harvest SCM repository (-cr) using Oracle (-o). This file also creates the tablespaces HARVESTMETA, HARVESTBLOB, and HARVESTINDEX. In addition, this file creates the database user's user name and password (-husr and -hpwd).

```
-o
-cr
-ausr=system
-apwd=manager
-husr=cascm
-hpwd=cascm
-svc=cascmsvc
-dvr=Oracle in OraDb10g_home1
-tsn=TEMP
-rbn=UNDOTBS1
-mfile=C:oracle\ora90\database\harvestmeta.ora
-bfile=C:\oracle\ora90\database\harvestblob.ora
-ifile=C:\oracle\ora90\database\harvestindex.ora
-msz=50
-bsz=50
-isz=50
```

Example: Configure the ODBC DSN, Create and Test the Repository, and Create and Test the Database User and Password (Oracle)

In this example, the sample response file configures the ODBC DSN, creates the CA Harvest SCM Oracle repository, and tests the repository (-tr). This file also creates the database user's name and password (-husr and -hpwd) and encrypts the database user's user name and password (-ep).

```
-o
-cr
-tr
-co
-ep
-ausr=system
-apwd=manager
-husr=cascm
-hpwd=cascm
-svc=cascmsvc
-dsn=cascm
-dvr=Oracle in OraDb10g_home1
-tsn=TEMP
-rbn=UNDOTBS1
-mfile=C:oracle\ora90\database\harvestmeta.ora
-bfile=C:\oracle\ora90\database\harvestblob.ora
-ifile=C:\oracle\ora90\database\harvestindex.ora
-msz=50
-bsz=50
-isz=50
```

How to Configure the Repository Using the hdbsetup Database Configuration Utility

Use the hdbsetup Database Configuration Utility to configure the CA Harvest SCM database and complete these steps:

- Create the repository
- Configure the ODBC DSN
- Create the database user (SQL Server only)
- Upgrade the repository
- Convert MDB to stand-alone schema (SQL Server Only)

  Important! This option deletes all the MDB tables that are not owned by CA Harvest SCM. If you are using the MDB with other products, do *not* use this option.

- Delete the repository
- Encrypt the database user's user name and password
- Test to see if a repository exists
- Load project lifecycle templates
- Load project XML form templates

How to Create the Repository (Oracle)

If you are using Oracle, use this option to create the CA Harvest SCM repository.

At the ENTER DESIRED ACTION prompt, enter CR.

In interactive mode, follow these steps:

1. Enter the service name:
   - For a local database server, press Enter to accept the blank service name.
   - For a remote database server, enter the service name that points to the remote server.
2. Enter the user name and password of the Oracle administrator.
3. Enter the user name and password of the product database user to be created. This user owns the product tables and has administrative access to the product repository being created.
4. Supply any additional data for which you are prompted, if applicable.
5. Confirm that all the values you entered are correct and then enter 0 or press Enter to continue.
6. Enter Y to continue or cancel the request.

After the repository is created, you are notified and are returned to the Configuration Menu.

To create the product repository on Oracle in command-line mode, use the following commands:

`-o -ausr=`*`username`* `-apwd=`*`password`* `-husr=`*`username`* `-hpwd=`*`password`* `-cr` *`options`*

- For -ausr and -apwd, *username* and *password* are the logon credentials of the Oracle administrator.
- For -husr and -hpwd, *username* and *password* are the logon credentials of the product user who owns the product tables and has administrative access to the product repository.
- *options* are the Oracle-only parameters that follow.
- Parameters

For the -mfile, -bfile, and -ifile parameters, the term “directory path” means the complete path name of the directory (but not the file names) of the tablespace files.

Note: For recommended values, see your Oracle documentation.

-rbn="*rollback-tablespace-name*"

Specifies the rollback tablespace name.

-tsn="*temp-tablespace-name*"

Specifies the temporary tablespace name.

-mfile=*fullname*

Specifies the file path and name for the meta tablespace file.

-msz=*size*

Specifies the initial size in MB for the meta tablespace file.

-bfile=*fullname*

Specifies the file path and name for the blob tablespace file.

-bsz=*size*

Specifies the initial size in MB for the blob tablespace file.

-ifile=*fullname*

Specifies the file path and name for the index tablespace file.

-isz=*size*

Specifies the initial size in MB for the index tablespace file.

-svc=*service-name*

Specifies the Oracle service name.

Create the Repository (SQL Server)

When you use the Database Configuration Utility and are using SQL Server, use this option to create the CA Harvest SCM repository.

Important! For SQL Server, if you have not already done so, verify that the Windows operating system user name you use to create the product database is granted the system administrator role in SQL Server. This requirement exists because the product database installation scripts use the SQL Server osql utility running with a trusted connection. For more information about how to grant this role, see your SQL Server documentation.

At the ENTER DESIRED ACTION prompt, enter CR.

In interactive mode, follow these steps:

1. Enter the server name:
   - For a local database server, press Enter to accept the default local server.
   - For a remote database server, enter the computer name of the remote server and the instance name. For example:

     Server Name=MachineName\InstanceName
2. Select an authentication mode for the database connection.
3. (If you select SQL Server authentication in Step 2) Enter the user name and password of the SQL Server administrator and the user name and password of the database user to be created. This user owns the product tables and has administrative access to the repository being created.
4. (If you select Windows authentication in Step 2) You do not need to perform any action in this step. The CA Harvest SCM database user for the database being created is assigned the same name as the Windows operating system user that you are currently logged in as. For example, if you are currently logged in as *admin*, the database user is named *admin* too. This user owns the tables and has administrative access to the repository being created.
5. Enter the database name.
6. Confirm that the values you entered are correct and then enter 0 or press Enter to continue.

   After the repository is created, you are notified and are returned to the Configuration Menu.

7. Use the following steps, based on your authentication mode, to create the repository.

SQL Server Authentication

To create the repository on SQL Server and use SQL Server authentication as the database connection method, use the following commands:

```
-s -cr -svr=servername -conmethod=Sqlserver -dbn databasename -ausr=username
-apwd=password -husr=username -hpwd=password
```

-ausr=username and -apwd=password

Specifies the user name and password of the SQL Server administrator.

-husr=*username* and -hpwd=*password*

Specifies the user name and password of the product user to be created. This user has administrative access to the database.

-svr=*servername*

Specifies the server name. The default is local.

-dbn=*dbname*

Specifies the SQL Server database name.

-conmethod=Sqlserver

Specifies SQL Server authentication as the database connection method.

Windows Authentication

To create the repository on SQL Server and use Windows authentication as the database connection method, using the following commands:

```
-s -cr -svr=servername -conmethod=Windows -dbn databasename
```

-svr=servername

Specifies the server name. The default is local.

-conmethod=Windows

Specifies Windows authentication as the database connection method.

-dbn=*dbname*

Specifies the SQL Server database name.

Configure the ODBC DSN (Data Source Name)

Configure your ODBC DSN (data source name) when you are prompted to do so or by entering CO at the ENTER DESIRED ACTION prompt.

In interactive mode, follow these steps:

1. Enter the data source name that you want CA Harvest SCM to use. If the DSN already exists, you will be prompted to confirm that you want to overwrite the existing DSN.
2. Enter the data source name.
3. Enter more information based on your DBMS:
   - Oracle users are prompted for the service name. The default is empty.
   - SQL Server users are prompted for the server name. The default is local.

   Typically, you configure your ODBC driver only once. However, if you set up a different database or server, you must update the DSN.

   Note: Before encrypting a database user's user name and password (-ep), verify that you have configured your ODBC driver (DSN).
4. For SQL Server only, you are prompted to specify a product database connection method: Windows authentication or SQL Server authentication.
5. Confirm all the information entered, verify that the values are correct, and then enter **0** or press Enter to continue.
6. Specify whether to test the new driver connection, enter **Y** for yes or **N** for no. If you specify **Y**, you may be prompted for DBMS administrator logon credentials, and you are informed whether the test succeeded or failed. You are then returned to the Configuration Menu.

The equivalent options for command line mode are as follows:

If you are using Oracle, enter the following options:

```
-0 -co -svc=alias -dsn=datasourcename
```

If you are using SQL Server, enter the following options to configure ODBC DSN:

```
-S -co -svr=servername -dsn=datasourcename -dbn databasename
```

For *servername*, the default is (local).

After the driver is configured, you are notified.

Note: If you use the Web Interface and you change the data source name at any time, record the new value and edit the Web Interface configuration file (Harweb.cfg) to specify the new value. This change is required to help ensure that the Web Interface and the product continue to connect to the same database.

Create a Database User

Important! When using Windows authentication, your Windows account is used for this operation. Verify that your Windows account user is granted the system administrator or security role in SQL Server.

When you use the Database Configuration Utility and are using SQL Server, use this option to create a CA Harvest SCM database user.

At the ENTER DESIRED ACTION prompt, enter CU.

Follow these steps:

1. Enter the server name (SQL Server) and the instance name, for example, S*erver Name=MachineName\InstanceName.*

   For a local database server, press Enter to accept the default local server.

   For a remote database server, enter the computer name of the remote server.
2. Enter the database name.
3. Select the authentication mode for the product database connection: Windows authentication or SQL Server authentication.
4. If you selected SQL Server authentication, enter the user name and password of the SQL Server administrator and the product database user.

   If you selected Windows authentication, enter the product user's Windows domain name and username when prompted.
5. Confirm that the values you entered are correct and then enter 0 or press Enter to continue.

After the repository is created, you are notified and are returned to the Configuration Menu.

In command line mode, use the following command to create a product database on SQL Server, specifying SQL Server authentication as the product database connection method.

```
-s -cu -svr=servername -conmethod=Sqlserver -dbn databasename -ausr=username
-apwd=password -husr=username -hpwd=password
```

-svr=*servername*

Specifies the server name (SQL Server).

-conmethod=Sqlserver

Specifies SQL Server authentication as the product database connection method.

-dbn=*databasename*

Specifies the SQL Server database name.

-ausr=*username* and -apwd=*password*

Specifies the user name and password of the DBMS administrator.

-husr=*username* and -hpwd=*password*

Specifies the user name and password of the product user to be created. This user has administrative access to the product database.

In command line mode, use the following command to create a CA Harvest SCM database user on SQL Server, specifying Windows authentication as the CA Harvest SCM database connection method.

```
-s -cu -svr=servername -conmethod=Windows -domain=domainname
```

-svr=*servername*

Specifies the server name (SQL Server).

-conmethod=Windows

Specifies Windows authentication as the product database connection method.

-domain=*domainname*

Defines the product database user's domain name or computer name.

How to Upgrade the Repository (Oracle)

If you are using Oracle and you are upgrading from an earlier release of CA Harvest SCM, this action is required.

To start the database upgrade, at the ENTER DESIRED ACTION prompt, enter UR.

Follow these steps:

1. Enter the service name used for Oracle.

   Note: You can leave the service name blank if the database is local and you only have one database service.

2. Enter the user name and password of the database administrator.

3. Enter the user name and password of the product user who owns the product tables and has administrative access to the database being updated.

4. Confirm that the values you entered are correct and then enter 0 or press Enter to continue.

In command-line mode, use the following syntax to upgrade the repository:

`-0 -ur -svc=`*`service-name`* `-ausr=`*`username`* `-apwd=`*`password`* `-husr=`*`username`* `-hpwd=`*`password`*
`-dvr i=`*`driver`* `-noprompt`

- For -ausr=*username* and -apwd=*password*, *username* and *password* are the logon credentials of the Oracle administrator.
- For -husr=*username* and -hpwd=*password*, *username* and *password* are the logon credentials of the product user who owns the product tables and has administrative access to the database.
- For -svc=*service-name*, *service-name* specifies the Oracle service name.
- For -dvr i=*driver*, *driver* specifies the name of the Oracle ODBC driver, for example, “Oracle in OraDb11g_home1”.

Upgrade the Repository (SQL Server)

If you are using SQL Server and you upgrading from an earlier release of CA Harvest SCM, this action is required.

Starting the Database Upgrade

At the ENTER DESIRED ACTION prompt, enter **UR**.

Enter Required Logon Credentials

In interactive mode, follow these steps:

1. Enter the server name for SQL Server.
2. Enter the database connection method: Windows or SQLserver.
3. (Only required if the database connection method is SQLServer) Enter the user name and password of an SQL Server administrator.
4. (Only required if the database connection method is SQLServer) Enter the user name and password of the product user who owns the product tables and has administrative access to the database being updated.
5. Confirm that the values you entered are correct and then enter **O** or press Enter to continue.

Command Line Options

In command line mode, use the following syntax to upgrade the repository:

```
-s -ur -svr=(local) -conmethod=Windows -dvr=driver -noprompt
```

or

```
-s -ur -svr=(local) -conmethod=Sqlserver -dvr=driver -ausr=username
-apwd=password -husr=username -hpwd=password -noprompt
```

- For -ausr=*username* and -apwd=*password*, *username* and *password* are the logon credentials of the SQL Server administrator.

  Note: The -ausr and -apwd are not required if you are using Windows authentication.

- For -husr=*username* and -hpwd=*password*, *username* and *password* are the logon credentials of the product user who owns the product tables and has administrative access to the database.

  Note: The -husr and -hpwd are not required if you are using Windows authentication.

- for -dvr=*driver*, *driver* specifies the name of the SQL Server ODBC driver, for example "SQL Native Client".

How to Delete the Repository (Oracle)

If you are using Oracle to delete (drop) the CA Harvest SCM repository, enter DR at the ENTER DESIRED ACTION prompt.

In interactive mode, follow these steps:

1. Enter the service name for Oracle.
2. Enter the user name and password of the Oracle administrator.
3. Enter the user name and password of the product user who owns the product tables and has administrative access to the product database being deleted.
4. Confirm all the information entered, verify that all the values are correct, and then enter **0**, or press Enter to continue.
5. Enter **Y** to confirm that you want to delete the database user, or enter **N** to cancel the operation.

   If you enter **N**, the operation is aborted, the database user remains, and the next step does not apply.

   If you enter **Y**, the database user is deleted, and you perform the next step.
6. Enter **Y** to confirm that you also want to delete the product tablespaces, or enter **N** to keep them.

You are notified that the operation is complete and are returned to the Configuration Menu.

Following are the equivalent options for command-line mode:

`-0 -dr -ausr` *`username`* `-apwd` *`password`* `-husr` *`username`* `-hpwd` *`password`* `-svc` *`servicename`* *`-dvr=driver`* `-noprompt`

- For -ausr *username* and -apwd *password*, *username* and *password* are the logon credentials of the Oracle database administrator.
- For -husr *username* and -hpwd *password*, *username* and *password* are the logon credentials of the product user who owns the product tables and has administrative access to the product database being deleted.
- For -svc *servicename*, *servicename* specifies the Oracle service name.
- For -dvr=*driver*, *driver* specifies the name of the Oracle ODBC driver, for example, “Oracle in OraDb11g_home1”.

How to Delete the Repository (SQL Server)

If you are using SQL Server, to delete (drop) the CA Harvest SCM repository, enter **DR** at the ENTER DESIRED ACTION prompt.

In interactive mode, follow these steps:

1. Enter the server name for SQL Server.
2. Enter the database connection method: Windows or SQLserver.
3. (Windows database connection only) Enter the user name and password of the SQL Server administrator.
4. (Windows database connection only) Enter the user name and password of the product user who owns the product tables and has administrative access to the product database being deleted.
5. (Windows database connection only) Enter the database name.
6. (SQL Server authentication only) Enter the username and password of the SQL ServerSQL Server administrator, and the database name.

   You are notified that the operation is complete and are returned to the Configuration Menu.

Use the following equivalent options for the command-line mode.

For SQLServer authentication:

```
-s -dr -svr=(local) -conmethod=Windows -dbn dbname -dvr=driver -noprompt
```

For database authentication:

```
-s -dr -svr=(local) -conmethod=Sqlserver -dbn dbname -ausr username -apwd password
-dvr=driver -prompt
```

- For -ausr *username* and -apwd *password*, *username* and *password* are the logon credentials of the SQL Server administrator.

  Note: The -ausr and -apwd are not required if you are using Windows authentication.

- For -dbn *dbname*, *dbname* specifies the SQL Server database name.
- For -dvr=*driver*, *driver* specifies the name of the SQL Server ODBC driver, for example, “SQL Native Client”.

Encrypt the Database User's User Name and Password

When you use the Database Configuration Utility and after the CA Harvest SCM database user's user name and password are created or updated, encrypt them by entering EP at the ENTER DESIRED ACTION prompt.

Important! Before encrypting the user name and password (-ep), verify that you have configured your ODBC driver (DSN).

In interactive mode, when prompted, enter the name and password of the product user who owns the tables and has administrative access to the database.

For SQL Server, if you use Windows authentication as the database connection method, you must perform this action (Encrypt Database Username and Password), but you do not need to enter a user name and password.

This action uses the svrenc utility with the -s option to encrypt the database user's user name and password for use by the product. Encrypting them is required for the product to connect to the database with this user name and password.

After the encryption is finished, you are notified and are returned to the Configuration Menu.

Important! The CA_SCM_HOME environment variable must be set to execute this command.

The equivalent options for command-line mode include the following:

Oracle

```
-O -ep -husr=username -hpwd=password
```

SQL Server with SQL Server authentication

```
-S -ep -conmethod=Sqlserver -husr=username -hpwd=password
```

SQL Server with Windows authentication

```
-S -ep -conmethod=Windows
```

All databases

-husr=*username* and -hpwd=*password*

Specifies the user name and password of the user who has administrative access to the database. For SQL Server using Windows authentication, specify the user name but not the password.

SQL Server

`-conmethod={Windows|Sqlserver}`

Specifies Windows or SQL Server authentication as the database connection method.

More information:

Configure the ODBC DSN (Data Source Name) (see page 240)
Create the CA Harvest SCM User and Default Directories (see page 115)

Verify that a Repository Exists

Use this option in the Database Configuration Utility to test whether a specific CA Harvest SCM repository exists.

In interactive mode, follow these steps:

At the ENTER DESIRED ACTION prompt, enter TR, and follow the instructions for your DBMS.

Follow these steps:

1. Enter the service name for Oracle. If your database server is on the local computer, press Enter to accept the default value.
2. If prompted, enter the user name and password of a DBMS administrator.
3. Enter the user name and password of the product user who has administrative access to the product database being tested.
4. When prompted, confirm that the values you entered are correct and then enter 0 or press Enter to continue.

Follow these steps:

1. Enter the server name for SQL Server. If your database server is on the local computer, press Enter to accept the default value.
2. Select the connection method for accessing the product database: Windows authentication or SQL Server authentication.

3. Enter the user name and password of the product user who has administrative access to the product database being tested. If you are using Windows authentication, skip this step.
4. When prompted, confirm that the values you entered are correct and then enter 0 or press Enter to continue.

   After the repository is tested, you are returned to the Configuration Menu. The results of the test appear. The results are recorded in the hdbsetup.log file in the CA_SCM_HOME\log directory.

The equivalent options for command line mode follow for each DBMS.

For Oracle:

`-0 -tr -svc=servicename -ausr=username -apwd=password -husr=username -hpwd=password`

For SQL Server, using SQL Server authentication:

`-S -tr -svr=servername -conmethod=Sqlserver -ausr=username -husr=username –dbn=databasename`

For SQL Server, using Windows authentication:

`-S -tr -svr=servername -conmethod=Windows –dbn=databasename`

For all databases:

`-ausr=username` and `-apwd=password`

Specifies the user name and password of the DBMS administrator.

`-husr=username` and `-hpwd=password`

Specifies the user name and password of the product user who has administrative access to the product database.

For Oracle only:

`-svc=servicename`

Defines the Oracle service name.

For SQL Server only:

`-svr=servername`

Specifies the server name.

`-conmethod={Windows|Sqlserver}`

Specifies Windows or SQL Server authentication as the product database connection method.

`-dbn=databasename`

Specifies the SQL Server database name.

Exit the Database Configuration Utility

To exit the utility, enter **X.** (X followed by a period) at the ENTER DESIRED ACTION prompt.

The Database Configuration Utility closes the command prompt window.

Set Security for eTrust CATop Secret

You can set security options for the remote z/OS agent. If you are using eTrust CA-Top Secret Security (eTrust CA-Top Secret), you must modify the eTrust CA-Top Secret Parameter File to include a new facility, AGNTD. Add the following commands to the Parameter File to create the facility.

```
FACILITY(USERn=NAME=AGNTD)
FACILITY(AGNTD=MODE=FAIL,ACTIVE)
FACILITY(AGNTD=PGM=AGNTD,NOABEND)
FACILITY(AGNTD=MULTIUSER,RES)
FACILITY(AGNTD=NOLUMSG,NOSTMSG)
FACILITY(AGNTD=NORNDPW)
TSS ADD(ALL) FAC(AGNTD)
FACILITY(AGNTD=NOPROMPT,NOAUDIT,NOTSOC)
```

In the first line, USER*n* represents the facility name (USER0-USER221); assign an unused facility name, such as USER222, to AGNTD.

Note: For more information about assigning facility names in the eTrust CA-Top Secret Parameter File, see your eTrust documentation.

Set Security for RACF

You can set Resource Access Control Facility (RACF) security settings for the remote z/OS agent.

Follow these steps:

1. Add RACF OMVS segments to set up OMVS users.

   The CA Harvest SCM z/OS agent runs as an HFS program in the z/OS UNIX System Services. All users logging into or starting the z/OS agent must have valid OMVS segments with valid group IDs. MVS data sets accessed through the z/OS agent must be protected by valid profiles.

   The multi-user agent runs as a USS daemon and impersonates users using getpwnam, _passwd, and seteuid functions. The agent requires additional RACF authority. For more information about establishing UNIX security, see IBM's *z/OS UNIX System Services Planning Guide*.

All users who use the z/OS agent must have valid OMVS segments with valid groups. Consider the following information:

- Use a unique UID for the OMVS segments. *Do not* use OMVSDFLT.
- OMVS segments must have valid group IDs.
- The following are valid RACF commands:

  `LU <userid> OMVS`

  This command lists the OMVS where the <userid> is the z/OS login ID.

  `ALTUSER OMVS NOUID`

  This command removes an existing UID.

  `ALTUSER OMVS AUTOUID`

  This command assign a unique UID to a specific user.

The following is a sample OMVS SEGMENT definition:

```
OMVS(UID(1) HOME(/U/userid) PROGRAM(/bin/sh))
```

2. Define an OMVS SEGMENT for both new and existing users using the ADDUSER or ALTUSER RACF Commands.

   The following is a sample OMVS SEGMENT definition:

   ```
   OMVS(UID(000000000) HOME(/U/userid) PROGRAM(/bin/sh))   [UID(000000000)
   indicates a “super user”]
   ```

3. Set up OMVS parmlib specifications. This is site specific and resides in SYS1.PARMLIB(BPXPRM00). An example of OMVS parmlib specifications follows:

```
MAXASSIZE(1073741824)
MAXPROCSYS(400)
MAXPROCUSER(200)
MAXUIDS(200)
MAXFILEPROC(1024)
MAXPTYS(512)
MAXTHREADTASKS(5000)
MAXTHREADS(10000)
MAXCPUTIME(7200)
SUPERUSER(OMVSKERN)
CTRACE(CTIBPX00)
VERSION('&UNIXVER.')
SYSPLEX(YES)
FILESYSTYPE TYPE(HFS)
         ENTRYPOINT(GFUAINIT)
FILESYSTYPE TYPE(UDS) ENTRYPOINT(BPXTUINT)
ROOT     FILESYSTEM('HFS.&SYSPLEX..ROOT')
         TYPE(HFS)
         MODE(RDWR)
MOUNT    FILESYSTEM('HFS.P390.HFS')
         TYPE(HFS)
         MODE(RDWR) NOAUTOMOVE
         MOUNTPOINT('/FX04')
MOUNT    FILESYSTEM('HFS.P390.TMP')
         TYPE(HFS)
         MODE(RDWR) NOAUTOMOVE
         MOUNTPOINT('/FX04/tmp')
MOUNT……
FILESYSTYPE TYPE(AUTOMNT) ENTRYPOINT(BPXTAMD)
FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM) ASNAME(ZFS)
FILESYSTYPE TYPE(NFS)
           ENTRYPOINT(GFSCINIT)
           ASNAME(NFSC)
FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP)
       TYPE(INET)
       ENTRYPOINT(EZBPFINI)
NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        MAXSOCKETS(30000)
        TYPE(INET)
        INADDRANYPORT(5555)
        INADDRANYCOUNT(1000)
NETWORK DOMAINNAME(AF_UNIX)
        DOMAINNUMBER(1)
        MAXSOCKETS(2000)
        TYPE(UDS)
```

4. Data set profiles may have to be defined to RACF. Similar to the OMVS parmlib previously described, this is site dependent.

Example: Assign a Protected Profile to the MVS Data Sets

The following sample commands will assign a protected profile to the MVS data sets that will be accessed through the remote agent. If this is not done, RACFIND will fail and pds members will not display.

```
SETROPTS GENERIC(DATASET) REFRESH
SETROPTS GENERIC(*)
SETROPTS GENERIC(DATASET)
ADDSD 'userid.*' FROM('SYS1.*') FCLASS(DATASET) FGENERIC GENERIC
```

After these settings are enabled, you can access MVS files using the Workbench Remote Agent, and successfully check elements in to the product.

Multi-User Agent Security

You can set security options for the remote z/OS agent. When setting up multi-user agent security, consider the following information:

- A super user must start the multi-user agent. All user IDs with OMVS segments that define a UID of zero are super users.

  Note: IBM recommends assigning BPX.SUPERUSER authority. For more information about assigning authority, see IBM's *z/OS UNIX System Services Planning Guide*. The users who start the multi-user agent must have *read* access to facility BPX.DAEMON.

- All executables called by the agent must be apf authorized and program controlled. Users must execute extattr +ap * in both the lib and bin directories in the agent installation directory. In addition, the C runtime library must be program controlled.
- Users assigned super user access through the BPX.SUPERUSER facility must execute the su command *before* starting the multi-user agent.

Use the following RACF commands as reference when setting up multi-user agent security:

```
RDEFINE FACILITY BPX.DAEMON UACC(NONE)
```

This command defines the facility for the daemon.

```
RDEFINE FACILITY BPX.SUPERUSER UACC(NONE)
```

This command defines the facility for the super user.

`SETROPTS CLASSACT(FACILITY)`

This command refreshes RACF.

`SETROPTS RACLIST(FACILITY)`

This command refreshes RACF.

`ALTUSER <username> OMVS (UID(nn) HOME('<homedir>' PROGRAM('bin/sh'))`

This command assigns access to the facility where the <username> is the user ID who starts the multi-user agent.

`PERMIT BPX.DAEMON CLASS(FACILTY) ID(<userid>) ACCESS(READ)`

This command assigns access to the facility where the <username> is the BPX.SUPERUSER who starts the multi-user agent.

`PERMIT BPX.FILEATTR.* CLASS(FACILITY)ID(<userid>) ACCESS(READ)`

This command grants access to set program control (for example, apf and shared) using the extattr command, where the <username> is the user ID who starts the multi-user agent.

`rl facility bpx.daemon authuser`

This command checks access to the bpx.daemon facility.

`rlist program *`

This command lists program control.

Troubleshooting the z/OS Agent Under RACF

Use the following information to help you troubleshoot the z/OS agent under RACF.

- Monitor the RACF log for violations.
- Be sure you can ping the broker from the z/OS computer.

Note: For more information about daemon setup and program control, see IBM's *z/OS UNIX System Services Planning Guide*.

Performance Improvement (Oracle)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can define a default service on any Oracle server.

Note: For information about supported Oracle releases, see the *Release Notes*. The most recent versions of the *Release Notes* and other product documentation is posted on http://ca.com/support http://www.ca.com/us/support.aspx.

Defining a default service enables co-resident ODBC applications (such as the product server) to connect to a default service while bypassing the SQL*NET and TCP/IP layers. As a result, the performance of these applications improves significantly.

On UNIX and Linux, to use a default service and thus connect to the local database, the shell environment must have environment variables set for ORACLE_HOME and ORACLE_SID. When the current settings for $ORACLE_HOME and $ORACLE_SID match one of the SID_DESC blocks found in the listener.ora file, a single open default service is defined for that $ORACLE_HOME-$ORACLE_SID pair. However, if no matching SID_DESC block exists in the listener.ora file, no open default service exists, and any attempt to make a local connection to the database fails.

Note: For an overview of backing up Oracle databases and other Oracle maintenance information, see the *Administrator Guide*. For detailed instructions to maintain Oracle databases, see your Oracle documentation.

Database Backup Information

Use the following references to find information about backing up the database:

- For an overview of backing up Oracle and SQL Server databases and other Oracle and SQL Server maintenance information, see the *Administrator Guide*.
- For detailed instructions to maintain Oracle databases, see your Oracle documentation.
- For detailed instructions to maintain SQL Server databases, see your SQL Server documentation.

Chapter 15: Configuring the Broker and Server

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

Broker and Server Communication

The CA Harvest SCM broker and server programs execute operations requested by product clients. The agent program acts as a file server to remote clients. The options for configuring these programs vary with the operating system.

The agent can be started as either a multi-user or single-user process.

- On UNIX and Linux, the privileged root user account is required to start the agent in multi-user mode.
- On Windows, the privileged user account named Administrator is required to act as part of the operating system.
- The single-user agent does not require a privileged user account for startup. However, the *same* logon credentials used to start the single-user agent *must* also be used to access it.

More information:

The Single-User Agent (see page 264)

How the Broker and Server Work (Windows)

Both the broker and server processes are executed as Windows programs. Like other programs, the broker and server processes are automatically terminated when the user who started them logs off the Windows system. The broker can also be started as a Windows service.

Start the Broker (Windows)

Both the broker and server (or servers) are started automatically after you select the Broker from the CA Harvest SCM program group. The broker can also be started directly from the command line by running the command **bkrd.exe**.

The hbroker.arg file located in the %CA_SCM_HOME% directory specifies the minimum and maximum number of product server processes to be started by the broker.

When you start the broker process from the Windows Start menu, a console window with the broker's name appears. This window confirms that the broker process was started. If the broker process starts successfully, the contents of the hbroker.arg file are displayed on the terminal window. Closing the broker terminal window shuts down the broker and server processes.

Check the log files for the broker and server processes if startup problems are encountered. You can find the log files in the following location:

- %*HOMEDRIVE*%%*HOMEPATH*%\.cascm\log
- %CA_SCM_HOME%\log (If the agent/broker is started as a service using Local System Account.)

Broker Options (Windows)

Use the start broker command to start the broker directly from the command line. You can set broker options by editing the hbroker.arg file located in the %CA_SCM_HOME% directory or by using the command line. Options set on the command line override those specified in the hbroker.arg file.

This command has the following format:

```
bkrd.exe [options]
```

Important! The option names for bkrd are case-sensitive. Verify that you enter these settings carefully and correctly.

-help

Displays the help file.

-verbose

Prints log messages on the window.

-homedir=*SCM home dir*

Specifies the server process location. This location must have /log subdirectory.

-homeserver [_*version*]=*home dir*

Specifies the location of the server's argument file and log files.

-rtserver=*computer name*

Connects to Enterprise Communicator (PEC) network on the computer.

-cmfile=*command file*

Parses a PEC command file.

-install[=*parent service*]

Installs the program in Windows service database.

-remove

Removes the program from Windows service database

-shutdown

Initiates shutdown on local computer (broker only).

The agent can also be shut down at the same time as the broker using the option -shutdown=all.

-dirserver[_*version*]= *dir*

Specifies the directory location of the server executable. You need to specify this argument when you have different versions of servers. If you do not specify the directory, the PATH environment variable is used.

-minserver[_*version*]= *int*

Specifies the minimum number of servers the broker starts, regardless of the waiting requests. If you do not specify a number, the broker starts one server.

-maxserver[_*version*]= *int*

Specifies the maximum number of servers the broker starts, regardless of the waiting requests.

-queuesize[_*version*]= *int*

Specifies the number of waiting requests in the queue before the broker starts a new server.

-killperiod=*second*

Specifies the idle period before the broker shuts down temporary servers.

-loadbalance=*level*

Specifies level of load balancing for multiple server computers-OFF, LEVEL1, LEVEL2.

-servermachine[_*version*] =*computer*

Specifies the name of the computer on which the servers are running.

-qtrace

Turns on the trace for the broker maintained queues and tables. When this option is specified, it directs the contents of the broker maintained queues and tables to the broker log file (or to the stdout if -verbose is specified) whenever the queues or tables are modified.

-trace

Displays PEC-related trace messages.

How the Broker Manages Server Processes (Windows)

You can use the -minserver, -maxserver, and -queuesize options to manage the server processes. At broker startup, a minimum number of servers are started (-minserver) and additional servers up to a maximum number (-maxserver) can be started as the level of activity increases. The queue size (-queuesize) determines how many user requests to put into queue (wait) before starting a temporary server process, that is, above the minimum value (-minserver).

For example, this simple scenario illustrates how the broker manages the server processes:

1. Start the broker directly from the command line by entering the command bkrd and use the following parameters: -minserver=1 -maxserver=2, and -queuesize=1.
2. Consider three users: User1, User2, and User3. When all three users are logged into the broker but are idle (for example, not checking out or sending any kind of request), no server processes are in use.
3. If all three users become active simultaneously (that is, check-out or check-in), three requests are generated. One request is allocated a server process immediately (-minserver=1). Another request is put into queue (-queuesize=1) until a server becomes available. The third request also goes into queue but results in the queue size being exceeded by one. Because the queue size is exceeded and the -maxserver option allows two servers, a temporary server is started to process the additional request. As a result, one user request remains in queue while two user requests are processed. As soon as one of the first two requests is finished processing, a server becomes available and is assigned to the remaining user request in queue.
4. After all user requests are finished processing, one server shuts down, returning the server count to the minimum value (-minserver=1). You can use the -killperiod option to set the server idle time limit (the period of inactivity after which the broker shuts down “temporary” servers).

Start the Server (Windows)

The administrator at each CA Harvest SCM site must determine how many server processes are required to meet the needs of their product users. Base this value on the maximum number of users who would simultaneously access the product at any time. In general, one product server process should be available for each product user.

The product broker automatically starts and manages all product server processes. The broker options -minserver and -maxserver (set in the hbroker.arg file) specify the minimum and maximum number of product server processes managed by the broker.

Important! Do not start the product server processes manually. The product broker is designed to manage the minimum and maximum number of product server processes dynamically.

Server Options (Windows)

You can set CA Harvest SCM server options by editing the HServer.arg file located in the %CA_SCM_HOME% directory.

Important! The option names for hserver are case-sensitive; therefore, verify that you enter these settings carefully and correctly.

-help

Displays the help file.

-verbose

Prints a log message on screen.

-homedir=*SCM home dir*

Specifies the server process location. This location must have /log subdirectory.

-rtserver=*computer name*

Connects to Enterprise Communicator network on the computer.

-auditenabled=*int*

Enables the server to record global audit policy events that are enabled in the global audit policy table (HARGLOBALAUDITPOLICY). 1= Enable (Default); 0=Disable.

-cmfile=*command file*

Parses a PEC command file.

-cm=[*i, a*]

Specifies the preferred connect method (to accept or initiate) to create the direct connections between the product server, client, and agent.

i initiates connection

a accepts connection

-install[=*parent service*]

Installs the program in Windows service database.

-remove

Removes the program from Windows service database.

-broker=*name or IP*

Specifies the computer on which the broker is running. The default is the local computer (-broker=local).

-version=*string*

Specifies the broker version; default = Default

-datasource=*string*

Specifies the ODBC data source.

-memorylimit=*megabytes*

Monitors the memory allocated and shut down (exit) when the -memorylimit level is reached. If the number of server processes falls below the minimum number defined in the hbroker.arg file, a new server process is started.

-enforcepatchversion=*flag*

Enforces the client and server to be at the same patch level.

0 means that the client can log in to a server at a different patch level. 1 means that the client cannot log in to a server that is at a different patch level. The default option setting is 0.

Note: Clients and servers at different patch levels may not operate correctly. It is recommended that your product server and client installations (and agent-only) have the same patch level.

-enforcepatchversionmsg=*customized error message for enforcepatchversion*

Appends a customized message to the standard error message when -enforcepatchversion=1 is specified in the HServer.arg file.

-enforcehiddenpswd=*flag*

Enforces hiding of passwords from command-line utilities.

0 means that hiding of passwords from command-line utilities is *not* enforced. 1 means that hiding of passwords from command-line utilities is enforced.

The default option setting is 0.

Note: On the startup of the product server or file agent process, if the enforcement of hiding of passwords on command lines is not turned on, a message will be printed to the product server or file agent log that this enforcement is disabled.

-enforcehiddenpswdmsg=*customized error message for enforcehiddenpswd*

Appends a customized message to the standard error message when enforcing the hiding of passwords from command-line utilities.

-saveremoteagentinfo

Allows product users to save login passwords to remote agent connections.

If this value is 0 or if no such argument exists in the HServer.arg file, all login passwords to remote agent connections in the registry on Windows platforms will be set to empty. The check box Save file agent info of the Connect to New Agent dialog will be disabled when product users try to log in to new agents in the Workbench.

If this value is 1, login passwords to remote agent connections will be saved in the registry on the Windows platform.

The check box Save file agent info of the Connect to New Agent dialog will be enabled. This lets CA Harvest SCM users decide to save or not to save passwords in the registry on Windows platforms.

-commsize=*int size in Kilobytes*

Specifies the size in Kilobytes of the PEC buffer for data sent between the agent and the server during check-in and check-out. For example, -commsize=128 means the buffer size will be set to 128 Kilobytes. The default size of this optional parameter is 63K. You can specify different sizes to determine the best result for check-in/check-out performance in your network environment.

-listsize=*int size of each database retrieve*

Specifies the number of records to retrieve from the database and send across the network at a time. For example, -listsize=250 means the server will retrieve 250 records, send them to the client, and wait for a confirmation response. The default size of this optional parameter is 200. You can specify different sizes to determine the best result for list processes like version selection performance in your network environment; however, you also need to consider the impact on the throughput for all users in the product.

-logging=*int level*

Writes detailed server information to the server log. Each log level displays the information of all levels below it. The levels are cumulative not mutually exclusive.

Level=1 displays ODBC errors and any error messages written by the relational database.

Level=2 displays the product Transaction name; the date and time it started; the date and time it ended; the name of the server and process ID that processed the transaction; and the duration of the transaction in milliseconds.

Level=3 displays the executed SQL statement, when it started, when it ended, the execution time in milliseconds, and the time in milliseconds since the last completed SQL.

Level=4 displays commit and rollback statements.

Level=5 displays the results of the SQL statements being executed.

-sdstageusername=user name

Specifies a user name of an internal user to be used for Stage Deploy operations. This parameter is only needed when externally defined users are using the CA Harvest SCM Software Delivery integration. Stage Deploy operations fail when invoked by externally defined users because the user's passwords are unavailable. This user must have proper access to the project. An admin user is a good candidate because this user name does not have to be known by anyone except the CA Harvest SCM Administrator.

-textsearchtimeout=*int value in minutes*

Specifies a time-out value in minutes for the version search function. The default value is no time-out. If you specify a negative value or zero, the value resets to no time-out.

-trace

Displays PEC-related trace messages.

More information:

The Server Logging Option (see page 331)

The Single-User Agent

The CA Harvest SCM agent lets you access file systems from the CA Harvest SCM client and perform check-in and check-out processes. For example, you can execute check-out processes to a remote UNIX or Linux file system from a Windows CA Harvest SCM client. The single-user agent does not require a privileged user account for startup. The user who starts it owns the single-user agent process; therefore, only the user who owns the single-user agent process can log in.

You can specify agent start options either on the command line or in the agent argument file, hagent.arg, located in the CA_SCM_HOME directory. The options in the agent argument file are available for the multi-user agent and the single-user agent. Options specified on the command line override the same options set in the hagent.arg file.

Compared to the previous release where the agent process used the rtserver parameter, the Release 12.5 agent process runs as an independent process listening at a particular port. The rtserver parameter is no longer required.

More information:

Agent Start Options (Windows) (see page 269)

Start a Single-User Agent

You can start a single-user agent by providing either the user name or the name of the encrypted file (dfo file) that has the user credentials.

Follow these steps:

1. Depending on whether you want to provide the user name or the dfo file, specify one of the following options either on the command line or in the hagent.arg file:

   User Name

   - agntd -usr=*username*

   Encrypted File

   - agntd –eh=*agentdfofilename*

The following considerations apply to the agent startup:

- On Windows, the user name is case-insensitive.
- On UNIX, Linux, and z/OS, the user account starting a single-user agent process must have write access to the $CA_SCM_HOME/log directory. If write access is not granted, a log file write error occurs.
- If the user name specified by the -usr option or or the username encrypted in the dfo file (provided with the –eh option) is not the same as the current user logged in to the host computer, an “incorrect user name” startup error occurs.
- If you are using the -eh option, you must have created the dfo file using the svrenc command. For more information about the svrenc command, see the *Command Line Reference Guide*.

Each user can start many single-user agent processes on one computer provided that they run on different ports, for example:

```
agntd –usr=John –port=6001
agntd –usr=John –port=6002
agntd –usr=John –port=6003
agntd –eh=agentdfo –port=6004

...
```

The same port number cannot be used to start single-user agents for different users on the same computer. For example:

```
agntd –usr=John1 –port=6000
agntd –usr=John2 –port=6000
...
```

If another user starts agent on the same port, error “port already in use” is displayed.

More information:

Agent Start Options (Windows) (see page 269)

Single-User Agent Password Options

The user starting the CA Harvest SCM agent process with the -usr option, can specify the single-user agent password or it can be automatically generated. The password specified is a session password defined at run time for this agent process. A different session password can be defined each time a single-user agent process is started.

Note: The agent password option does not apply when you use the encrypted file with the -eh option.

The user starting the agent can specify the password with or without using the -pwd option:

- The -pwd option can be specified in the hagent.arg file or on the command line to explicitly set the single-user agent session password.
- If the -pwd option is not specified, you are prompted for a session password (specifying -pwdmethod=prompt is the same as not specifying -pwd).

The single-user agent password can be automatically generated using the -pwdmethod=random option. This option generates a random session password and displays it to the standard output device (terminal by default).

Example

```
C:\>agntd -usr=John -pwdmethod=random
CA SCM Agent Generated Password: 4zQBcNmX
Creating windowless child process ...
Parent process exiting ...
C:\>
```

Single-User Agent Log File

An agent log file is generated on startup in the CA_SCM_HOME\log directory. The naming format is as follows:

```
YyyymmddHAgentPID_UserName_PortNumber.log
```

PID

Specifies the agent process ID.

PortNumber

Specifies the value of the –port option.

An hco Command Line Example

The check-out command line utility (hco) supports connecting to a remote agent. For example, you can use the hco command with a remote agent connection to perform a check-out to a remote file system.

The following example shows a possible syntax for a remote agent check-out. The hco command is run from a Windows client to check out to a UNIX or Linux file system.

Note: The port number is the required parameter to specify for the agent.

The following example shows a possible syntax for a remote agent check-out. The hco command is run from a Windows client to check out to a UNIX or Linux file system:

```
hco -b “win01” -en “Production” -st “Dev” -vp “\ProdRep” -cp “/users/application” -br
-op as -s “*” -usr “myself” -pw “mypassword” -rm “sparc” -rusr “hartest” -rpw “test”
–rport 5000 -o “c:\Application\log\hco.log”
```

Note: For detailed descriptions of the CA Harvest SCM command-line options, see the *Command Line Reference Guide*.

How to Shut Down the Single-User Agent

The agent can be shut down by using the following command:

```
agntd –usr=username –port=portNumber –shutdown
```

The single-user agent can be shut down using an idle timeout option specified at agent startup, -timeout=*integer*. *integer* is the agent idle time limit in minutes. When the idle timeout limit is reached, the agent process shuts down automatically.

Start a Multi-User Agent (Windows)

You can start a multi-user CA Harvest SCM agent by selecting Agent from the CA Harvest SCM program group or by entering **agntd** on the command line, followed by any applicable startup options. You can start the multi-user agent as a service or a stand-alone process.

The port number on which the agent should run must also be specified either on the command line or in the HAgent.arg file. The agent will not start unless the port number is specified.

When an agent is started, it uses one thread. For each user who logs in, a new thread is spawned to handle the connection. When a user logs out, the thread for that connection terminates.

Follow these steps:

1. On the agent computer, click Settings, Control Panel, Administrative Tools, Local Security Settings.
2. The Local Security Settings window appears.
3. Click Local Policies to expand it and select User Rights Assignment.

   Policies and security settings are listed.
4. Add the login user to each of the following local security policies:
   - Act as part of the operating system
   - Replace a process level token
5. Restart the system.

A multi-user file agent may fail to start because a file agent service is already running. If this occurs, you may want to stop and remove the file agent service.

Note: If the multi-user agent is started as a service, you do not need to perform the previous steps, because the agent is started as a Service Account that has special privileges.

Follow these steps:

1. Stop the service if necessary.
2. Run agntd with the -remove option to remove the service from the Service Control Manager (SCM).
3. Start the file agent as a regular application (started directly from a desktop window, Startup menu, and so on).

More information:

Agent Start Options (Windows) (see page 269)

Agent Start Options (Windows)

The following CA Harvest SCM agent start options are available for both the multi-user agent and the single-user agent. You can specify the agent start options in the HAgent.arg file located in the %CA_SCM_HOME% directory or from the command line. If the –port option is not specified (either on the command line or in the HAgent.arg file), the agent fails to start. The multi-user agent creates a log file named *Yyyymmdd*HAgentPID_UserName_PortNumber.log in the %CA_SCM_HOME%\log directory.

The agntd command has the following format:

```
agntd [options]
```

Important! The option names for agntd are case-sensitive. Verify that you enter these settings carefully and correctly.

Example: Start a multi-user agent

```
agntd -port=5000
```

-port

Specifies the port that the agent listens to.

Note: For those options that require you to specify a file location, the Universal Naming Convention (UNC) is not supported for specifying client paths. Instead of entering the UNC location *server**share**directory**file*, you must enter *drive*:*directory**file*.

The following agent options are available for both the multi-user agent and the single-user agent.

-help

Displays the help file.

-verbose

Prints a log message on the window.

-homedir=*SCM home dir*

Specifies the agent location. This location must have /log subdirectory.

-port=*portnumber*

Specifies the port number that the agent listens to.

-cmfile=*command file*

Parses a PEC command file.

-cm=[*i , a*]

Specifies the preferred connect method (to accept or initiate) to create the direct connections between the product's server, client, and agent.

i initiates connection

a accepts connection

-install[=*parent service*]

Installs the program in Windows service database. A single-user file agent cannot be run as a service on Windows.

-remove

Removes the program from Windows service database.

-usr

Specifies your system login name (single-user mode only).

-pwd

Specifies your password (single-user mode only).

-eh

Specifies the dfo filename that contains the encrypted single user agent username and password (single-user mode only). You must create the dfo file using the svrenc command with the –f option. For more information about the svrenc command, see the *Command Line Reference Guide*.

Note: The -eh option is mutually exclusive with the -usr option.

-shutdown

Terminates the specified agent on the local computer.

Note: The agent shutdown can be controlled by the broker shutdown. The broker shutdown option, -shutdown=all, shuts down all agent processes that are visible in the RTserver network.

-lockdir=*lock directory*

Specifies that agntd is permitted to access only this directory and its subdirectories.

Note: Relative lock directories (for example, ../../user01/CA SCM) cannot be specified. If the product detects a relative -lockdir specification, the file agent will not start.

-timeout=*int*

Specifies the agent idle timeout in minutes. When the idle timeout is reached, the agent shuts down (single-user mode only).

-pwdmethod=*string*

Allows users to use one of two methods, “prompt” or “random,” for establishing the single-user agent password.
If -pwdmethod=prompt, the default method, the agent will prompt the user at start up to enter a password.
If -pwdmethod=random, the agent will display the randomly generated password that must be used to log in to the agent.

-enforcepatchversion=*flag*

Enforces the client and agent to be at the same patch level. "0" means that the client can log in to an agent at a different patch level. "1" means that the client cannot log in to an agent that is at a different patch level.

The default option setting is "0".

Note: Clients and agents at different patch levels may not operate correctly. It is recommended that your client and agent installations have the same patch level.

-enforcepatchversionmsg=*customized error message for enforcepatchversion*

Appends a customized message to the standard error message when "-enforcepatchversion=1" is specified in the HAgent.arg file.

-enforcehiddenpswd =*flag*

Enforces hiding of passwords from command-line utilities.

"0" means that hiding of passwords from command-line utilities is *not* enforced. "1" means that hiding of passwords from command-line utilities is enforced.

The default option setting is "0".

Note: On the startup of the server or file agent process, if the enforcement of hiding of passwords on command lines is not turned on, a message will be printed to the server or file agent log that this enforcement is disabled.

-enforcehiddenpswdmsg=*customized error message for enforcehiddenpswd*

Appends a customized message to the standard error message when enforcing the hiding of passwords from command-line utilities.

-trlvl=*int*

Specifies a trace mode or no trace. The value of *int* must be 0, 1, 2, or 3. If an *int* value is not specified or if you specify a value not in the range, the trace turns off automatically. Each trlvl mode is distinct and does not include information from other trace modes.

0 specifies no trace.

1 traces direct connect creation, thread creation, and login operations.

2 traces file input/output operations and only displays if a function related to file input/output fails.

3 traces the invocation of hexecp.

-trace

Display PEC-related trace messages.

More information:

The Agent Trace Facility (see page 330)

z/OS Agent Start Options

When the z/OS agent is started with the homedir argument and the value of homedir is the high-level qualifier (HLQ) of the Hagent argument file, the z/OS agent uses MVS data sets for the agent log file and the agent argument file. The agent argument file must exist with the following name: [assign the value for hlq in your book].HAGENT.ARG, where [assign the value for hlq in your book] is the high-level qualifier specified for the homedir argument. A sequential file will be allocated for the log file with the following name: [assign the value for hlq in your book].L1yymmdd.HAGENT, where [assign the value for hlq in your book] is the high-level qualifier specified on the homedir argument, yymmdd is the two digit year, month, and day, respectively.

The HAgent.arg file is a sequential file located under the $CA_SCM_HOME folder. By default, this file contains the name of the broker specified when the agent was installed, this name is specified by the -rtserver flag.

The following information summaries the z/OS agent start options. The options are available for both the multi-user agent and the single-user agent.

Important! The option names for agntd are case-sensitive; therefore, verify that you enter these settings carefully and correctly.

-help

Displays this message to SYSPRINT.

-cm=[*i* , *a*]

Specifies the preferred connect method (to accept or initiate) to create the direct connections between the product server, client, and agent.

i initiates connection

a accepts connection

-usr

Specifies the system login name.

-pwd

Specifies the password for the system login name.

-homedir=*SCM home dir*

Specifies the agent location. This location must have /log subdirectory. To allocate MVS data sets for the argument and log files, specify the high-level qualifier of the agent argument file.

-port=*portnumber*

Specifies the port number that the agent listens to.

-timeout=*int*

Specifies the agent idle timeout in minutes. When the idle timeout is reached, the agent shuts down (single-user mode only).

-lockdir=*lock directory*

Specifies that agntd only has access to this directory and its subdirectories.

Note: Relative lock directories (for example, ../../user01/caSCM) cannot be specified. If the product detects a relative -lockdir specification, the file agent will not start.

-pwdmethod=*string*

Allows users to use one of two methods, “prompt” or “random,” for establishing the single-user agent password.
If -pwdmethod=prompt, the default method, the agent will prompt the user at start up to enter a password.
If -pwdmethod=random, the agent will display the randomly generated password that must be used to log in to the agent.

-shutdown

Terminates the specified agent on the local computer.

Note: The product agent shutdown can be controlled by the product broker shutdown. The broker shutdown option, -shutdown=all, shuts down all product agent processes that are visible in the RTserver network.

-enforcepatchversion=flag

Enforces the client and agent to be at the same patch level. "0" means that the client can log in to an agent at a different patch level. "1" means that the client cannot log in to an agent that is at a different patch level.

The default option setting is "0".

Note: Clients and agents at different patch levels may not operate correctly. It is recommended that your product client and agent installations have the same patch level.

-enforcepatchversionmsg=*customized error message for enforcepatchversion*

Appends a customized message to the standard error message when "-enforcepatchversion=1" is specified in the HServer.arg file.

-enforcehiddenpswd =*flag*

To enforce hiding of passwords from command-line utilities.

"0" means that hiding of passwords from command-line utilities is *not* enforced. "1" means that hiding of passwords from command-line utilities is enforced.

The default option setting is "0".

Note: On the startup of the product server or file agent process, if the enforcement of hiding of passwords on command lines is not turned on, a message will be printed to the product server or file agent log that this enforcement is disabled.

-enforcehiddenpswdmsg=*customized error message for enforcehiddenpswd*

A customized message that is appended to the standard error message when enforcing the hiding of passwords from command-line utilities.

-trlvl=*int*

Specifies a trace mode or no trace. The value of *int* must be 0, 1, 2, or 3. If an *int* value is not specified or if you specify a value not in the range, the trace turns off automatically. Each trlvl mode is distinct and does not include information from other trace modes.

0 specifies no trace.

1 traces direct connect creation, thread creation, and login operations.

2 traces file input/output operations and only displays if a function related to file input/output fails.

3 traces the invocation of hexecp.

-trace

Display PEC-related trace messages.

More information:

The Agent Trace Facility (see page 330)

The USS User ID

To execute a single-user agent, a valid USS user ID must be defined. The user ID must have a unique uid assigned to the user name such that the whoami command returns the login name. If the user ID is defined with the USS default uid, an error occurs when the user attempts to start the agent. The user must also have READ access to the following IBM Facility: BPX.DAEMON.

To start the multi-user agent, a user must have a valid USS user ID that can act as root. The user must be able to execute the extattr command to APF authorize the agent executable and the dlls. READ access to the following IBM Facilities is required:

```
BPX.DAEMON
BPX.FILEATTR.APF
BPX.FILEATTR.PROGCTL
BPX.SUPERUSER
```

For eTrust CA-Top Secret, the following commands must be applied to a user ID to start a multi-user agent:

```
TSS PER(userid) IBMFAC(BPX.DAEMON) ACCESS(READ)
TSS PER(userid) IBMFAC(BPX.FILEATTR.APF ) ACCESS(READ)
TSS PER(userid) IBMFAC(BPX.FILEATTR.PROGCTL) ACCESS(READ)
TSS PER(userid) IBMFAC(BPX.SUPERUSER ) ACCESS(READ)
```

How Client and Server Patch Levels are Enforced

The HAgent.arg -enforcepatchversion option forces the login process to fail in the following cases:

- If an agntd process with the -enforcepatchversion set to "1" is used on the server and if the client version number (for example, 5.2, 5.2.1) does not match the server's value, or if the client uses a version of the haragent library that does not send the client's version/patch.
- Continuing with the previous case, if the versions match but the patch values differ (or if the client uses a version of the haragent library that does not send version/patch information).

How Hidden Passwords are Enforced

The HAgent.arg -enforcehiddenpswd option enforces the hiding of passwords from command line utilitiescommand-line utilities allow for the specification of CA Harvest SCM user credentials or user credentials for a remote operating system (for example, for a file agent login).

If the server or file agent process does not allow command-line utilities to pass clear-text passwords and if the command line is determined to be passing clear-text passwords, then the server or file agent process will reject the create session or file agent login request.

Password information will be considered hidden on command-line utilities if any of the following are true:

- The command line specifies one of the input file options (for example, -i or -di).
- The input file option is not specified and the password option (for example, -pw, -rpw) is not specified on the command line.

How to Start the Broker as a Service

The CA Harvest SCM broker can be installed and started as a Windows service. When the program runs as a service, it sends output that you typically see when you run it as -verbose. The output is sent to log files in your %CA_SCM_HOME%\log directory. The log file name includes the current date as a suffix. Each time the broker service is started, a new log file is created for the product broker and server processes.

If the database server resides on the *same* host as the product server, follow these steps:

1. Enter the following command from a command prompt:

   `bkrd.exe -install=`*`parent-service`*

   where *parent-service* is the database service name for the DBMS you are using:

   - *(Oracle)* Specify OracleService*instance*, where *instance* is the name of your Oracle instance. For example, if the Oracle instance is named ORCL, enter the following command:

     ```
     bkrd.exe -install=OracleServiceORCL
     ```

   - *(SQL Server 2005)* Specify SQL Server (MSSQL$Instance), where *instance* is the name of your SQL Server instance. For example, for a default installation, enter the following command:

     ```
     bkrd.exe -install=SQL Server (MSSQLSERVER)
     ```

2. To start the broker service, restart the system.

If the database server is on a *different* host than the product server, follow these steps to install the product broker as a service:

1. Enter the following from a command prompt:

   ```
   bkrd.exe -install
   ```

2. To start the broker service, restart the system.

To remove the broker service

1. Run the following command:

   ```
   bkrd -remove
   ```

2. Restart the system.

To disable the broker service

1. Open Services in the Windows Control Panel.
2. On the Services window, select CASCM Broker Service and click Startup.
3. Select Disabled and click OK.
4. Close the windows.

Start the Multi-user Agent as a Service

The CA Harvest SCM multi-user agent can be started as a Windows service when the operating system is started. When the program runs as a service, it sends output that you typically see when you run it as -verbose. The output is sent to log files in your %CA_SCM_HOME%\log directory. The log file name includes the current date as a suffix. Each time the multi-user agent service is started, a new log file is created.

Follow these steps:

1. Run the following command:

   ```
   agntd.exe -install
   ```

2. Open the HAgent.arg file and locate the following line, and specify a number:

   ```
   -port=portnumber
   ```

3. Save your changes and close the HAgent.arg file.
4. To start the agent service, restart the system.

Follow these steps:

1. Run the following command:

   ```
   agntd -remove
   ```

2. Restart the system.

Follow these steps:

1. Open Services in the Windows Control Panel.
2. On the Services window, select CASCM Agent Service and click Startup.
3. Select Disabled and click OK.
4. Close the windows.

Connect to a Broker Using a Remote RTserver

By default, a CA Harvest SCM client connects to a broker using an RTserver located on the same computer as the broker. You can configure a product client to use the same remote RTserver as the broker (that is, the broker is running on one computer and the RTserver is located on a different computer) so that the client can successfully execute the command-line programs.

To connect to a broker using a remote RTserver, specify the *-rtserver=RTserver_name* option in the hclient.arg file on the client computer. If this option is not specified, the product assumes that the RTserver is running on the broker computer.

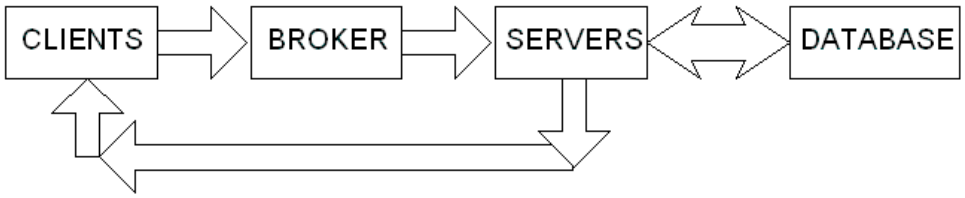
How a Broker Manages Multiple Servers on Multiple Computers

On Windows computers, you can set up the broker to manage servers on its local computer, remote computers, or both. An unlimited number of remote computers can be specified in the hbroker.arg file to be used for running servers. The remote server computers must have an agent running (started by a user with administrative, Act as operating-system privileges). The owners of these agent processes username and password must be encrypted in the broker's password file.

A quick review of the transaction process follows:

1. Client computers send requests to the broker.
2. The broker relays each request to a particular server.
3. The server communicates with the database as needed and sends the results directly back to the client.

The following illustration shows clients communicating with a broker, and servers communicating with a database and clients.



As previously illustrated, Multiple clients and servers follow this path of clients to broker to servers to database.

The broker can disperse the client requests to the server in two ways, as specified by the loadbalance=[level1|level2] line in the hbroker.arg file:

- loadbalance=level1 means that requests to the different server computers are dispersed in a fixed cyclic order. For example, given two CA Harvest SCM server computers, requests would be made in the following pattern: computer1-computer2-computer1...
- loadbalance=level2 means that the broker sends the request to one of the idle servers on the computer with the largest number of inactive servers. Ties are broken arbitrarily.

To set up your computers to use one broker, one database, and multiple servers and clients, do the following:

1. Configure each server host computer.
2. Using your DBMS networking utilities, configure the server computers to access the DBMS.
3. Initialize the DBMS to support maximum user needs.

   Add the following line to the rtserver.cm file on the broker computer:

   `setopt server_names` *`computer1`*`,` *`computer2`*`,` *`computer3`*`…`

   where *computer#* is the name of the computer that you want to start servers on.

   (If a line in the rtserver.cm file contains the text “setopt server_names” append the computer names to this line.)
4. Start an agent on each server computer.
5. Run the program bkrdenc to add the encrypted username and password to the broker's list. On the broker computer, enter the following command at the command prompt:

   `bkrdenc -m` *`computer_name`* `-usr` *`username`* `-pw` *`password`*
6. Execute this command once for each server computer.

   For help and for a list of other options, enter the following command:

   `bkrd -help`

7. Modify the hbroker.arg file on each server computer to specify how many servers the broker starts on each computer and the type of load balancing that is required. The following sample hbroker.arg file specifies multiple users on multiple computers using one broker and one database.

```
//Global to all server computers
-loadbalance=LEVEL2
-verbose

//This is the local computer
-minserver=5
-maxserver=10
-queuesize=5
-homeserver=C:\Program Files\CA\SCM
-dirserver=C:\Program Files\CA\SCM

// Mary's computer
-servermachine_mary=maryscomputer
-minserver_mary=20
-maxserver_mary=100
-queuesize_mary=3
-homeserver_mary=C:\Program Files\CA\SCM
-dirserver_mary=C:\Program Files\CA\SCM

// John's computer
-servermachine_john=johnscomputer
-minserver_john=30
-maxserver_john=200
-queuesize_john=3
// Notice in this case -homeserver_john is not the CA_SCM_HOME location.
-homeserver_john=D:\Program Files\John
-dirserver_john=C:\Program Files\CA\SCM
```

This setup starts 5 servers on the broker computer, 20 servers on Mary's computer, and 30 servers on John's computer when the broker is started. John and Mary both have their own broker versions denoted by the *_version* notation, for example, _john.

Also no version is specified for the broker computer. When no version is specified, the version name “Default” is used. You may also want to add a version to the local computer, to maintain the naming convention.

No version name, including “Default,” can be used more than once.

Start the Broker (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can start the broker directly from the command line by running the command **bkrd**. You can modify the HBroker.arg file located in the $CA_SCM_HOME to specify the minimum and maximum number of CA Harvest SCM server processes to be started by the broker. Only one broker, but multiple servers, can be run at a time per host computer.

Follow these steps:

1. Enter the following command to move to the directory where the product is installed:

   ```
   cd $CA_SCM_HOME
   ```

2. Enter the following command to move to the /bin directory:

   ```
   cd bin
   ```

3. Enter the following command:

   ```
   ./bkrd -option
   ```

Broker Options (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Use the start broker command to start the broker directly from the command line. You can set broker options by editing the HBroker.arg file located in the $CA_SCM_HOME% directory or by using the command line. Options set on the command line override those specified in the HBroker.arg file.

This command has the following format:

```
bkrd [options]
```

Important! The option names for bkrd are case-sensitive. Verify that you enter these settings carefully and correctly.

-help

Displays this message.

-verbose

Prints log message on the window.

-homedir=*SCM home dir*

Specifies the server process location. This location must have /log subdirectory.

-homeserver [_*version*]=*home dir*

Specifies the location of the server's argument file and log files.

-rtserver=*computer name*

Connects to the PEC network on the computer.

-cmfile=*command file*

Parses a PEC command file.

-shutdown

Initiates shutdown on local computer (broker only).

The agent can also be shut down at the same time as the broker using the option -shutdown=all.

-dirserver[_version]=*dir*

Specifies the directory location of the server executable. You need to specify this argument when you have different versions of servers. If you do not specify the directory, the PATH environment variable is used.

-minserver[_version]=*int*

Specifies the minimum number of servers the broker starts, regardless of the waiting requests. If you do not specify a value, the broker starts one server.

-maxserver[_version]=*int*

Specifies the maximum number of servers the broker starts, regardless of the waiting requests.

-queuesize[_version]=*int*

Specifies the number of waiting requests in the queue before the broker starts a new server.

-killperiod=*second*

Specifies the idle period before the broker shuts down temporary servers.

-loadbalance=*level*

Specifies the level of load balancing for multiple server computers-OFF, LEVEL1, LEVEL2.

-servermachine[_*version*] =*computer*

Specifies the computer name that the servers are running on.

-umask=*a three digit octal number*

Controls log file permissions. If not set to a valid octal digit, this value is ignored without an error message, and log files have read/write permissions for all users.

-qtrace

Turns on the trace for the broker maintained queues and tables. When this option is specified, it directs the contents of the broker maintained queues and tables to the broker log file (or to the stdout if -verbose is specified) whenever the queues or tables are modified.

-trace

Displays PEC-related trace messages.

How the Broker Manages Server Processes (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can use the -minserver, -maxserver, and -queuesize options to manage the server processes. At broker startup, a minimum number of servers are started (-minserver) and additional servers up to a maximum number (-maxserver) can be started as the user activity increases. The queuesize (-queuesize) determines how many user requests to put into queue (wait) before starting a temporary server process, that is, above the minimum value (-minserver).

For example, this simple scenario illustrates how the broker manages the server processes:

1. The broker is started from the command line or from the program group, using the following parameters: -minserver=1-maxserver=2 and -queuesize=1.
2. Consider three users, User1, User2 and User3. When all three users are logged into the broker but “idle” (for example, not checking out or sending any kind of request), no server processes are in use.
3. If all three users become active simultaneously (that is, check-out or check-in), three requests are generated. One request is allocated a server process immediately (-minserver=1). Another request is put into queue (-queuesize=1) until a server becomes available. The third request also goes into queue but results in the queue size being exceeded by one. Because the queue size is exceeded and the -maxserver option allows two servers, a temporary server is started to process the additional request. As a result, one user request remains in queue while two user requests are processed. As soon as one of the first two requests is finished processing, a server becomes available and is assigned to the remaining user request in queue.
4. After all user requests are finished processing, one server shuts down, returning the server count to the minimum value (-minserver=1). You can use the -killperiod option to set the server idle time limit (the period of inactivity after which the broker shuts down “temporary” servers).

Start the Server (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The CA Harvest SCM broker automatically starts and manages all product server processes. Administrators at each site must determine how many server processes are required to meet the needs of their product users. Base this value on the maximum number of users who would simultaneously access the product at any time. In general, one product server process should be available for each product user. The broker options -minserver and -maxserver (set in the HBroker.arg file) specify the minimum and maximum number of product server processes managed by the broker.

Important! Do not start the product server processes manually. The product broker is designed to manage the minimum and maximum number of product server processes dynamically.

Server Options (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can set the server options by editing the HServer.arg file located in the $CA_SCM_HOME directory.

Important! The option names for hserver are case-sensitive; therefore, verify that you enter these settings carefully and correctly.

-help

Displays this message

-verbose

Prints log message on screen

-homedir=*SCM home dir*

Specifies the server process location. This location must have /log subdirectory.

-rtserver=*computer name*

Connects to the PEC network on the computer

-cmfile=*command file*

Parses a PEC command file

-cm=[*i , a*]

Specifies the preferred connect method (to accept or initiate) to create the direct connections between the CA Harvest SCM server, client, and agent.

i initiates connection

a accepts connection

-broker=*Name or IP*

The computer the broker is running on, default = local

-version=*string*

Specifies the broker version, default = Default

-datasource=*string*

Specifies the ODBC data source

-memorylimit=*value in megabytes*

Monitors the memory allocated and shut down (exit) when memorylimit level is reached. If the number of server processes falls below the minimum number defined in the HBroker.arg file, a new server process is started.

-enforcepatchversion=*flag*

Enforces the client and server to be at the same patch level. "0" means that the client can log in to a server at a different patch level. "1" means that the client cannot log in to a server that is at a different patch level.

The default option setting is "0".

Note: Clients and servers at different patch levels may not operate correctly. It is recommended that your product server and client installations (and agent-only) have the same patch level.

-enforcepatchversionmsg=*customized error message for enforcepatchversion*

Appends a customized message to the standard error message when "-enforcepatchversion=1" is specified in the HServer.arg file.

Note: The text you enter after the equal sign displays to users exactly as you enter it. Quotation marks are *not* required in the message text. If you enclose the message text in quotation marks, the quotation marks are displayed to users when the message is issued.

-enforcehiddenpswd =*flag*

Enforces hiding of passwords from command-line utilities.

"0" means that hiding of passwords from command-line utilities is *not* enforced. "1" means that hiding of passwords from command-line utilities is enforced.

The default option setting is "0".

Note: On the startup of the product server or file agent process, if the enforcement of hiding of passwords on command lines is not turned on, a message will be printed to the product server or file agent log that this enforcement is disabled.

-enforcehiddenpswdmsg=*customized error message for enforcehiddenpswd*

Appends a customized message to the standard error message when enforcing the hiding of passwords from command-line utilities.

-saveremoteagentinfo

Allows product users to save login passwords to remote agent connections.

If this value is 0 or if no such argument exists in the HServer.arg file, all login passwords to remote agent connections in the registry on Windows platforms will be set to empty. The check box “Save file agent info” of the “Connect to New Agent” dialog will be disabled when product users try to log in to new agents in the Workbench.

If this value is 1, login passwords to remote agent connections will be saved in the registry on the Windows platform.

The check box “Save file agent info” of the “Connect to New Agent” dialog will be enabled. This allows product users to decide to save or not to save passwords in the registry on Windows platforms.

-commsize=*int size in Kilobytes*

Specifies the size in Kilobytes of the PEC buffer for data sent between the agent and the server during check-in and check-out. For example, -commsize=128 means the buffer size will be set to 128 Kilobytes. The default size of this optional parameter is 63K. You can specify different sizes to determine the best result for check-in and check-out performance in your network environment.

-listsize=*int size of each database retrieve*

Specifies the number of records to retrieve from the database and send across the network at a time. For example, -listsize=250 means the server will retrieve 250 records, send them to the client, and wait for a confirmation response. The default size of this optional parameter is 200. You can specify different sizes to determine the best result for list processes like version selection performance in your network environment; however, you also need to consider the impact on the throughput for all users in the product.

-logging=*int level*

Writes detailed server information to the server log. Each log level displays the information of all levels below it. The levels are cumulative not mutually exclusive.

Level=1 displays ODBC errors and any error messages written by the relational database.

Level=2 displays the product Transaction name; the date and time it started; the date and time it ended; the name of the server and process ID that processed the transaction; and the duration of the transaction in milliseconds.

Level=3 displays the executed SQL statement, when it started, when it ended, the execution time in milliseconds, and the time in milliseconds since the last completed SQL.

Level=4 displays commit and rollback statements.

Level=5 displays the results of the SQL statements being executed.

-umask=*a three digit octal number*

Controls log file permissions. If not set to a valid octal digit, this value is ignored without an error message, and log files have read/write permissions for all users.

-sdstageusername=user name

Specifies a user name of an internal user to be used for Stage Deploy operations. This parameter is only needed when externally defined users are using the CA Harvest SCM Software Delivery integration. Stage Deploy operations fail when invoked by externally defined users because the user's passwords are unavailable. This user must have proper access to the project. An admin user is a good candidate because this user name does not have to be known by anyone except the CA Harvest SCM Administrator.

-textsearchtimeout=*int value in minutes*

Specifies a time-out value in minutes for the version search function. The default value is no time-out. If you specify a negative value or zero, the value resets to no time-out.

-trace

Displays PEC-related trace messages.

Start a Multi-User Agent (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The agent can be started as a multi-user process with options specified from the command line, or you can specify the agent start options in the HAgent.arg file located in the $CA_SCM_HOME directory. The agent cannot be started unless a port number is specified.

Follow these steps:

1. Enter the following command to become the root user:

   ```
   su root
   ```

2. Move to the $CA_SCM_HOME/bin directory by entering the following command:

   ```
   cd $CA_SCM_HOME/bin
   ```

3. Enter the following command:

   ```
   ./agntd -option
   ```

4. The multi-user agent creates a log file in the $CA_SCM_HOME/log directory. The format is:

   *Yyyymmdd*HAgentPID_UserName_PortNumber.log

Example: Starting a multi-user agent

```
./agntd -port=5000
```

-port

Specifies the port number that the agent listens to.

The previous example can be set up so the -port option is specified in the HAgent.arg file as follows:

```
-port=5000
-verbose
```

Agent Options (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The following CA Harvest SCM agent start options are available for both the multi-user agent and the single-user agent.

Important! The option names for agntd are case-sensitive. Verify that you enter these settings carefully and correctly.

-help

Displays the help content to start and use the agent.

-verbose

Prints log message on the window.

-homedir=*SCM home dir*

Specifies the agent location. This location must have /log subdirectory.

-port=*port number*

Connects to the port on which the agent is running.

-cmfile=*command file*

Parses a PEC command file.

-cm=[*i , a*]

Specifies the preferred connect method (to accept or initiate) to create the direct connections between the product's server, client, and agent. The following options are available:

i

Initiates connection

a

Accepts connection

-usr

Specifies the system login name (single-user mode only).

-pwd

Specifies the password (single-user mode only).

-eh

Specifies the dfo filename that contains the encrypted single user agent username and password (single-user mode only). You must create the dfo file using the svrenc command with the –f option. For more information about the svrenc command, see the *Command Line Reference Guide*.

Note: The -eh option is mutually exclusive with the -usr option.

-shutdown

Terminates the specified agent on the local computer.

Note: The agent shutdown can be controlled by the broker shutdown. The broker shutdown option, -shutdown=all, shuts down all agent processes that are visible in the RTserver network.

-timeout=*int*

Specifies the agent idle timeout in minutes. When the idle timeout is reached, the agent shuts down (single-user mode only).

-lockdir=lock directory

Specifies that agntd only has access to this directory and its subdirectories.

Note: Relative lock directories (for example, ../../user01/CA SCM) cannot be specified. If the product detects a relative -lockdir specification, the file agent will not start.

-pwdmethod=*string*

Allows users to use one of two methods, “prompt” or “random,” for establishing the single-user agent password.

- If -pwdmethod=prompt, the default method, the agent will prompt the user at start up to enter a password.
- If -pwdmethod=random, the agent will display the randomly generated password that must be used to log in to the agent.

-enforcepatchversion=*flag*

Enforces the client and agent to be at the same patch level. The following option settings are available:

- (Default) "0" means that the client can log in to an agent at a different patch level.
- "1" means that the client cannot log in to an agent that is at a different patch level.

Note: Clients and agents at different patch levels may not operate correctly. It is recommended that your client and agent installations have the same patch level.

-enforcepatchversionmsg=*customized error message for enforcepatchversion*

Appends a customized message to the standard error message when "-enforcepatchversion=1" is specified in the HAgent.arg file.

-enforcehiddenpswd=*flag*

To enforce hiding of passwords from command-line utilities.

"0" means that hiding of passwords from command-line utilities is *not* enforced. "1" means that hiding of passwords from command-line utilities is enforced.

The default option setting is "0".

Note: On the startup of the server or file agent process, if the enforcement of hiding of passwords on command lines is not turned on, a message will be printed to the server or file agent log that this enforcement is disabled.

-enforcehiddenpswdmsg=*customized error message for enforcehiddenpswd*

Appends a customized message to the standard error message when enforcing the hiding of passwords from command-line utilities.

-umask=*a three digit octal number*

Controls log file permissions. If not set to a valid octal digit, this value is ignored without an error message, and log files have read/write permissions for all users.

On UNIX/Linux environments, the shell's umask value affects the file permission bits for check-out files and directories, CA Harvest SCM daemon logs (Broker, Server, and Agent logs), and signature files. Multi-user agents require a umask value to clear the UNIX/Linux group's write bit (for example, a umask octal value of 002 instead of a typical default 022) where the Shared Working Directory option is used at check-out. Prior to starting a multi-user agent, the UNIX/Linux shell umask needs to be set to clear the group's write bit to allow users to check-out and replace read-only files that are not owned. After you change the umask value and restart the agent, signature files and subdirectories for existing check-out paths will not have their bits updated, only signature files and subdirectories on newly created check-out paths will be updated. Check-outs may still fail on already existing check-out paths, unless the group's write bit is manually cleared for the path's subdirectories and signature files.

Note: For more information about the shared working directory option, see the *Administrator Guide*.

-trlvl=*int*

Specifies a trace mode or no trace. The value of *int* must be 0, 1, 2, or 3. If an *int* value is not specified or if you specify a value not in the range, the trace turns off automatically. Each trlvl mode is distinct and does not include information from other trace modes.

0

Specifies no trace.

1

Traces direct connect creation, thread creation, and login operations.

2

Traces file input/output operations and only displays if a function related to file input/output fails.

3

Traces the invocation of hexecp.

-trace

Display PEC-related trace messages.

Important! If you are using LDAP authentication, configure the agent to use LDAP authentication.

More information:

External Authentication Configuration (see page 297)
The Agent Trace Facility (see page 330)

OS System-Related Settings and Agent Operations

UNIX system-related resource limit settings, including maximum threads per process, file descriptor limit, and thread stack memory, may impact the agent operations.

The Maximum Threads Allowed for a Process and Computer

The maximum number of threads per process is a UNIX kernel parameter, which can be queried using the following syntax:

On HPUX:

```
# hp: /usr/bin/kmtune
```

On HP-UX, the kernel parameter 'max_thread_proc' specifies the maximum number of threads a process can create. On some older systems the value of this kernel parameter may need to be raised and would subsequently require you to rebuild the kernel. Consult with your system administrator regarding the kernel parameter settings.

On Solaris or AIX, the following syntax can be used to query kernel parameters:

```
# solaris: /usr/ccs/bin/nm /kernel/genunix
# aix: check /usr/include/unistd.h or /etc/lsattr -E -l sys0
```

Because the CA Harvest SCM agent creates threads on demand, this particular kernel parameter will affect how many threads the agent can create, which in turn will affect how many concurrent transactions an agntd process can have. For each check-out, the product agent will create two threads:

- One for the direct connection between the check-out client and the product agent
- One for the direct connection between the product agent and the product server

Note: For information about how to change the kernel parameters, refer to the UNIX computer man page or vendor documentation.

The Number of System Open File Descriptors for a Process

Executing a large number of concurrent check-out processes involves the use of many file descriptors in the agent process. The user needs to set this value appropriately; otherwise, you will experience a “too many open files” error. For check-out transactions, the file descriptor consumption is a linear progression on the number of concurrent check-out operations.

In the agent process, it will require a minimum of seven file descriptors for STDIN, STDOUT, STDERR, Rtclient to RTserver socket, and agent listening port file descriptor. With each check-out to the remote agent operation, five additional file descriptors are opened:

- Two for the direct connection socket file descriptors: CA Harvest SCM client to agent and product agent to server
- One for the product signature file
- One writes to the actual file being checked out
- One for the temporary file used in the decompression operation

The product agent is a multi-threaded program. When a user runs multiple concurrent check-outs, multiple threads are created and they access different files simultaneously. Depending on the number of CPUs available on the agent computer and number of concurrent operations as well as an appropriate file descriptor setting will be required.

On UNIX systems, the command 'ulimit -a' reports the current file size resource limitation for the processes started under your login shell. Some of these resource limits are user configurable and the values associated with them are known as the 'soft limits.' The administrator can set the corresponding 'hard limit' values for the resources at the system level.

On Solaris systems, 'ulimit -n', shows the maximum number of file descriptors available to your login shell. For example, a user can set the soft limit on the number of file descriptors by using the command:

```
ulimit  -n 1024
```

If needed, the system administrator can raise or set the file descriptor limitation for the entire system by specifying the 'rlim_fd_max' (hard limit) and 'rlim_fd_cur' (soft limit) by adding them to the /etc/system file. For example in the following case, the hard limit for the number of file descriptors is being set to 8192 and the soft limit is set to 1024:

```
set rlim_fd_max=8192
set rlim_fd_cur=1024
```

In addition, system calls used by the product agent require open file descriptors. On Sun Solaris, the studio library limits open file descriptors to 256 per process for 32-bit applications. The agent process will retry a failed attempt to get an open file descriptor; otherwise, the “open too many files” error will return to the user. An example of such an error in the hco log file follows:

```
I00060040: New connection with Broker xyzxyz  established.
Attempting to log into remote agent...
Agent login failed (error code = -24). <Too many files opened>
```

If a user encounters this file descriptor shortage error, the user needs to rerun the product operation.

Note: For more information about file size limitation, refer to the ulimit man page on UNIX platforms.

Additional Computer Resource-Related Limitations

UNIX systems may be configured differently based on usage. Usually, the system kernel parameters are set to reflect the limit on the resources that the users can claim for the processes run by their login shell. The default values for the system resources on the older systems may no longer be sufficient for proper usage of multi-threaded or highly concurrent applications.

Like any other multi-threaded application, the CA Harvest SCM agent requires a certain amount of system resources for its proper operation. You may need to raise the limit on file descriptors, number of threads per process, and the limit on the stack size depending on the number of simultaneous users using the agent and the type of operations being carried out by those users. For example, the kernel parameter 'maxdsiz' for 32 - bit HP is at a 1GB limit.

Depending on the operating system implementation, when the thread is created it requires either a pre-allocated or pre-reserved stack for each thread. Some operating systems commit that memory stack and others do not. Therefore, the number of threads that the agent can create are also determined by the user stack limit, the kernel stack, and the data limits.

External Authentication Configuration

hDuring the CA Harvest SCM installation, you specify the method your site will use to authenticate users' names and passwords when they attempt to log in to the product: internal (CA Harvest SCM) authentication or external authentication. External authentication uses an authentication server to validate users' credentials (user names and passwords); for example, a Lightweight Directory Access Protocol (LDAP) v3-compliant LDAP server, such as Novell eDirectory. Optionally, you can configure the CA Harvest SCM server to use mixed-mode authentication where you use an authentication server for authentication of most user accounts, but lets the CA Harvest SCM user administrator create and maintain additional internally authenticated users for CA Harvest SCM.

After installing the product and the Web Interface, you can reconfigure them to use a different authentication mode than the one specified during the installation. You can configure multi-user remote agents, servers, and Web Interface servers to use either internal or external authentication to validate user names and passwords when users attempt to log in to the product.

External Authentication

For CA Harvest SCM clients and users, external authentication provides a seamless step to the login process. When a product client connects to a product remote agent or server, the client must supply login credentials (the user name and password).

When external authentication is enabled, the product remote agent or server connects to the authentication server on start-up (for example, an LDAP server). If it fails to connect to the authentication server, it exits and records the failure of the connection attempt in the standard message log.

When using external authentication, product users are required to explicitly provide user name and password to log in to the product. This requirement applies even if the authentication server used by the product is the same one used to authenticate operating system credentials. Additionally, the user name and password used to log in to the product can be different than the ones used to log in to the operating system.

LDAP Authentication

In a Microsoft Windows environment, LDAP authentication is supported in a single domain only; it is not supported across multiple trusted domains.

Secure communication between a CA Harvest SCM LDAP client (a product remote agent or server) and an LDAP server is supported with Transport Layer Security (TLS) and Secure Socket Layer (SSL) and requires the use of certificates.

Users are the only type of external identity used by the product. User groups defined in the external authentication servers are not used by the product. The product recognizes only the user groups and user group memberships that are created and maintained in the product.

When external authentication is enabled, users cannot access the product unless they are defined with the same user name in both the product and the external authentication server. Therefore, after enabling external authentication, administrators must add or rename user names in the product to match the corresponding user names in the authentication server. When external authentication is enabled, any user name not defined in the external authentication server cannot access the product.

The Authentication Server

If you are using OpenLDAP with TLS or SSL encryption mode, to enhance security, you can optionally create the following files on the external authentication server:

- The trusted certificate file
- The client certificate and private key files

Note: To enable security on your external authentication server, see your system administrator or your external authentication server documentation.

More information:

The OpenLDAP Certificate and Key Files for TLS or SSL Encryption Mode (see page 315)

How to Enable External Authentication on the Multi-User Remote Agent

When external authentication is enabled on the remote agent, login credentials provided to the remote agent are validated against the external authentication server.

Follow these steps:

1. Shut down the multi-user remote agent.
2. Set up the external authentication configuration and connection options in the HAgent.arg file for the authentication mode you are using (for example, authmode=openldap).
3. Restart the remote agent.

How To Enable External Authentication on the Broker, Server, and Web Interface

When external authentication is enabled on the CA Harvest SCM broker, server, and Web Interface, login credentials provided to the broker are validated against the external authentication server.

Important! Verify that at least one administrator user (member of the product Administrators group) exists in the external authentication server at all times.

Note: The initial product user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of 1. This user is always a product administrator and always exists in the product, even if this user does not exist in the external authentication server. However, when you use external authentication, this user (like all other product users) must exist in the external authentication server to log in to the product.

To enable external authentication on a broker and its connected servers and Web Interface servers, do the following:

1. Shut down the broker and the servers and Web Interface servers connected to it (the broker). For example, if broker 1 is connected to servers A and B and Web Interface servers A and B, shut down all of these components.
2. Set up the external authentication configuration options for the broker in the HBroker.arg file for the authentication mode you are using (for example, openldap).
3. Set up the external authentication configuration and connection options for the server in the HServer.arg files for the authentication mode you are using.
4. If you are using the Web Interface, set up the external authentication configuration options for the Web Interface in the Harweb.cfg files for the authentication mode you are using.

5. Set up the value of the CASESENSLOGIN field in the HARTABLEINFO table in the database to match the behavior of the external authentication server.

   For both internal and external authentication, during login, create user, and update user operations:

   - If the value of the CASESENSLOGIN field in the HARTABLEINFO table is N, the user name authentication is not case-sensitive.
   - If the value of the CASESENSLOGIN field in the HARTABLEINFO table is Y, the user name authentication is case-sensitive (this is the default setting after a server installation).

6. Restart the broker and its connected servers and Web Interface servers.
7. (Optional) Change the product user names to match the corresponding user names in the external authentication server.

Important! When external authentication is enabled, users cannot access the product unless they (the users) are defined with the same user name in both the product and the external authentication server.

How to Disable External Authentication on the Multi-User Remote Agent

When external authentication is disabled on the multi-user remote agent, login credentials provided to the remote agent are validated against the operating system.

Follow these steps:

1. Shut down the remote agent.
2. Set the authentication mode for the agent to internal by doing one of the following in the agent's HAgent.arg file:
   - Set -authmode=internal
   - Delete the -authmode=setting
3. (Optional) Delete all external authentication configuration and connection options from HAgent.arg file.

   Note: When authentication mode is internal, these options are ignored.
4. Restart the remote agent.

How To Disable External Authentication on the Broker, Server, and Web Interface

When external authentication is disabled on the CA Harvest SCM broker, server, and Web Interface, login credentials provided to the broker are validated against the internal user data.

Follow these steps:

1. Shut down the broker and the servers and Web Interface servers connected to it (the broker). For example, if broker 1 is connected to servers A and B and Web Interface servers A and B, shut down all of these components.
2. Set the authentication mode for the broker to internal by editing the broker's HBroker.arg file and doing all of the following actions that apply:
   - Set -authmode=internal
   - Set -authmode[_*version*]=internal
   - Delete the -authmode= setting
   - Delete the -authmode[_*version*]= setting
3. (Optional) Delete all external authentication configuration options in the HBroker.arg file.

   Note: When authentication mode is internal, these options are ignored.

   Set the authentication mode for the servers to internal by doing one of the following in each server's HServer.arg file:
   - Set -authmode=internal
   - Delete the -authmode= setting
4. (Optional) Delete all external authentication configuration and connection options from HServer.arg file.

   Note: When authentication mode is internal, these options are ignored.
5. Set the authentication mode for the Web Interface servers to internal by doing one of the following in each Web Interface server's Harweb.cfg file:
   - Set AuthMode=internal
   - Delete the AuthMode=setting

6. Restart the broker and its connected servers and Web Interface servers.

   Important! The next step is optional but is *very* important for security reasons.

   (Optional) Do the following:

   a. Set the Password Policy according to your security guidelines.

   b. Set new passwords for all users.

   c. Force “Change Password on Next Logon” for all users.

Note: The initial user created during the installation is identified by the record in the HARUSER table whose USROBJID field has a value of 1. This user is always a product administrator and always exists in the product. This user's password retains its original value in the product (the value at the time the servers switched to external authentication).

Multi-User Remote Agent Options

Use the following options to configure external authentication for CA Harvest SCM multi-user remote agents that have external authentication enabled. Enter or modify these options in the remote agent's HAgent.arg file.

-authmode={openldap|internal}

Important! An individual remote agent's authentication mode is independent of, and can be different than, the authentication mode used by the product brokers, servers and other remote agents.

(Optional) Specifies the authentication mode used by the product multi-user remote agent to authenticate users' logon credentials:

internal

Uses internal (CA Harvest SCM) authentication.

openldap

Uses an LDAP v3-compliant directory server (LDAP server).

Default: internal

external authentication connection options

Defines the connection options for the authentication mode you are using.

Broker Options

Use broker options to configure external authentication for CA Harvest SCM brokers that have external authentication enabled. Enter or modify these options in the broker's HBroker.arg file.

-authmode[_*version*]={openldap|internal}

(Optional) Specifies the authentication mode used by the product broker.

Important! The value defined for the product broker must also be used by the product servers and the Web Interface servers connected to it. However, this value is independent of, and can be different than, the value used by other product brokers.

internal

Uses internal (CA Harvest SCM) authentication.

openldap

Uses an LDAP v3-compliant directory server (LDAP server).

Default: internal

-authsynchinterval[_*version*]=*dd*[:*hh*[:*mm*[:*ss*]]]

Defines the authentication synchronization interval between the product broker and the authentication server. Use the input format *dd*[:*hh*[:*mm*[:*ss*]]], where *dd* is days, *hh* is hours, *mm* is minutes, and *ss* is seconds.

Default: 1 (1 day)

Minimum: 0:1 (1 hour)

Note: If the value of the authentication synchronization interval is invalid or less than one hour, the broker uses the minimum value (1 hour).

Limits: 20 characters

Examples:

-authsynchinterval=1:4 specifies 28 hours (1 day plus 4 hours).

-authsynchinterval=1:4:6 specifies 28 hours plus 6 minutes (1 day plus 4 hours plus 6 minutes).

-authsynchinterval=0:4:0:30 specifies 4 hours plus 30 seconds.

Note: The broker sends an authentication synchronization request to an available product server at the following times: once when the product broker is started and each time that the broker's authentication synchronization time interval has been reached. For product users that exist in the external authentication server, the synchronization refreshes the values of the following CA Harvest SCM user attributes: Real Name, Phone Number, Phone Extension, Fax Number, and eMail. For CA Harvest SCM users that do not exist in the external authentication server, the users are ignored and their properties are not updated.

When authentication synchronization completes, the broker log shows a summary message indicating the number of product users ignored and updated. For example:

```
External authentication synchronization summary: Users ignored: 2;
Users updated: 5.
```

Server Options

Use the following options to configure external authentication for CA Harvest SCM servers that have external authentication enabled. Enter or modify these options in the server's HServer.arg file.

-authmode={openldap|internal}

(Optional) Specifies the authentication mode used by the product server.

Important! The value for the server must be the same as the value used by its product broker.

internal

Uses internal (CA Harvest SCM) authentication.

openldap

Uses an LDAP v3-compliant directory server (LDAP server).

Default: internal

external authentication connection options

Defines the connection options for the authentication mode you are using.

mixedauthmode={1|0}

(Optional) Specifies whether the CA Harvest SCM server uses mixed-mode authentication mode.

1

Allows internal (CA Harvest SCM) accounts to be maintained and authenticated.

0

All accounts are external.

Default: N

Web Interface Options

Use the following options to configure external authentication for the Web Interface that have external authentication enabled. Enter or modify these options in Harweb.cfg, the Web Interface configuration file.

AuthMode={openldap|internal}

(Optional) Specifies the authentication mode used by the Web Interface server.

Important! The value for the Web Interface server must be the same as the value used by its CA Harvest SCM broker.

internal

Uses internal (CA Harvest SCM) authentication.

openldap

Uses an LDAP v3-compliant directory server (LDAP server).

Default: internal

MixedAuthMode={1|0}

(Optional) Specifies whether the CA Harvest SCM server uses mixed-mode authentication mode.

1

Allows internal (CA Harvest SCM) accounts to be maintained and authenticated.

0

Specifies all accounts to be external.

Default: N

OpenLDAP External Authentication Connection Options (Servers and Remote Agents)

Use the following options to configure the LDAP connection options for CA Harvest SCM servers and remote agents that have external authentication enabled (for example: authmode=openldap). These connection options apply to both the server and agent. Enter or modify these options in the server's HServer.arg file or the agent's HAgent.arg file.

Important! Always enclose a value in quotation marks (" ") when it contains spaces.

-ldapserver=”*hostname1*[:*port1*] [*hostname2*[:*port2*] [*hostname3*[:*port3*]…]”

Defines one or more host names of the LDAP server, for example:

```
-ldapserver=hostname1
```

You can optionally define the port number to use on each host, by entering the host name in the form **hostname:port**, for example:

```
-ldapserver=hostname2:389
```

You can specify a list of host names separated by spaces and enclosed in quotation marks. Each host may optionally be of the form **hostname:port**, for example:

```
-ldapserver=”hostname1:389 hostname2 hostname3:389”
```

Important! If used, the *:port* option overrides the port number provided in the -ldapport parameter.

Limits: 255 characters

When the -ldapserver=*hostname*[:*port*] parameter defines multiple host names, the product server or agent connects to the first available LDAP server in the list.

-ldapport=*portnumber*

Specifies the port number for the LDAP server computer. This parameter is used if the LDAP port number is not specified in the -ldapserver=*hostname*[:*port*] parameter.

Default: If -ldapmode=ssl, then the default is 636; otherwise, the default is 389.

Minimum: 1

Maximum: 9999

-ldapbinddn=distinguished-name

Defines the LDAP initial bind distinguished name (DN) to the LDAP Server. For all authentication operations, only the initial DN is used to bind to the LDAP directory. A sample entry is:

```
-ldapbinddn="cn=john22,ou=users,ou=north america,dc=abccorp,dc=com"
```

Default: None

Limits: 255 characters

-ldapbindpw=*password*

Defines the password for the LDAP distinguished name used for the initial bind. Do *not* enter spaces.

If you do not specify a password, an empty password is used.

Important! The value you specify is saved in clear text. For security reasons, it is recommended that you provide the password using the -ldapbindpwfile=*{filename}* parameter.

Limits: 255 alphanumeric characters

-ldapbindpwfile=*filename*

Specify either the file name or the complete path name for the file containing the encrypted password.

Default: If you do not specify a complete path name, the default path name is used. The default path name is the product installation directory: %CA_SCM_HOME% on Windows or $CA_SCM_HOME for UNIX and Linux.

If the -ldapbindpwfile=*filename* parameter is specified, it overrides the value provided in -ldapbindpw=*password*.

Note: Use the command line utility svrenc to generate a file containing encrypted username and password values. The password value is the only one used; the username value is ignored. For more information, see the *Command Line Reference Guide*.

-ldapfilter=*search-filter*

Defines an RFC-2254-compliant search filter for locating a user. For example, when a user attempts to log in to product, this filter is used to search for the user in the LDAP server.

Default: (&(objectclass=person)(*user-attribute-name*=<*placeholder*>))

Note: The complete expression for the search filter used by your LDAP server may differ from the default value, depending on how your LDAP server has been configured. For details, see your system administrator.

(user-attribute-name=<placeholder>)

Specifies the LDAP User attribute name and its placeholder used in the search.

user-attribute-name

Defines your LDAP server's attribute name for user name. This value *must* be the same as the value specified for your LDAP server by the LDAP User Attribute name parameter, -ldapattrusrname=*attribute name*.

<placeholder>

Identifies a literal constant placeholder for *user-attribute-name*. Enter exactly the same value as *user-attribute-name* and enclose the value with angle brackets (< >), as shown in the following examples.

Examples

These examples use the default search filter.

If -ldapattrusrname=uid for your LDAP server, then the search filter is:

(&(objectclass=person)(uid=<uid>))

If -ldapattrusrname=cn for your LDAP server, then the search filter is:

(&(objectclass=person)(cn=<cn>))

If -ldapattrusrname=uname for your LDAP server, then the search filter is:

(&(objectclass=person)(uname=<uname>))

Examples: How the Search Filter is Used

The search filter is used to find a user name when it is required by any operation. For example, consider (&(objectclass=person)(uid=<uid>)): When a user attempts to log in to the product, <uid> is replaced dynamically with the user's user name, and the LDAP directory is searched for this user.

These examples use the default search filter and use the setting -ldapattrusrname=uid:

When the user amy33 attempts to log on, the search filter used to locate this user is:

(&(objectclass=person)(uid=<amy33>))

When the user john22 attempts to log on, the search filter used to locate this user is:

(&(objectclass=person)(uid=<john22>))

-ldapsearchtimeout=*seconds*

(Optional) Defines the number of seconds to search for a user in the LDAP directory; for example, when a user attempts to log in to the product.

Default: 60 (1 minute)

Limits: 20 digits

-ldapbasedn=*base-distinguished-name*

Defines the base distinguished name (DN) used when searching in the LDAP server. For example:

```
-ldapbasedn="ou=users,ou=north america,dc=abccorp,dc=com”
```

Default: None

Limits: 255 characters

-ldapmode={none|tls|ssl}

Specifies the security mechanism to use for authenticating product users:

tls

Specifies Transport Layer Security.

Specify TLS *only* if your LDAP server supports StartTLS.

ssl

Specifies Secure Socket Layer.

none

Specifies no security mechanism.

Important! If you specify none (no encryption), user credentials and all other information exchanged between the product and the LDAP server is transmitted in clear-text mode.

Default: tls

-tlstrcertfile=*filename*

(Optional) Defines the complete path name of the TLS trusted certificate file. This parameter specifies the PEM-format file containing certificates for the Certificate Authorities (CAs) that the LDAP client (the product remote agent or server) will trust. The certificate for the CA that signed the LDAP server certificate must be included in these certificates. If the signing CA was not a top-level (root) CA, certificates for the entire sequence of CAs from the signing CA to the top-level CA should be present. Multiple certificates are simply appended to the file; the order is not significant.

You can also define the TLS trusted certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CACERT *filename*

-tlscertfile=*filename*

(Optional) Defines the complete path name of the TLS client certificate file.

You can also define this certificate file in the OpenLDAP configuration file (for example: on UNIX, in $HOME/.ldaprc file) using the following parameter:

TLS_CERT *filename*

-tlskeyfile=*filename*

(Optional) Defines the complete path name of the TLS private key associated with the client certificate file.

You can also define this key in the OpenLDAP configuration file (for example: on UNIX, in the $HOME/.ldaprc file) using the following parameter:

TLS_KEY *filename*

Important! Private keys themselves are sensitive data and are usually password-encrypted for protection. However, the current LDAP API implementation does not support encrypted keys. Therefore, the key must not be encrypted and the file containing the key must be protected carefully.

-ldapattrusrname=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for user name.

Examples include -ldapattrusrname=cn and -ldapattrusrname=uname.

OpenLDAP External Authentication Connection Options (Remote Agents, UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The following options apply *only* to CA Harvest SCM remote agents on UNIX and Linux.

Enter or modify these options in the agent's HAgent.arg file.

Important! This section describes the *UNIX-user* attributes, in other words, the attributes of a UNIX user. The UNIX-user attributes are UNIX user id, UNIX primary group id, UNIX home directory, and UNIX shell.

When you are using LDAP authentication, during a login to the product's remote agent, after login credentials are validated against the LDAP server, the remote agent attempts to obtain the UNIX-user attributes from the LDAP server.

If the UNIX user ID and UNIX primary group ID are not available from the LDAP server, the remote agent then attempts to get the UNIX-user attributes from the local operating system, using the user name provided in login credentials. If the remote agent is unable to obtain the UNIX-user attributes, the login attempt to the remote agent fails.

UNIX accounts on the LDAP server must be represented by the RFC 2307 posixAccount object class to enable the remote agent to use POSIX account attributes. Therefore, it may be necessary to extend your LDAP schema to support POSIX account attributes.

Note: For details on how to set up "POSIX account attributes" on your LDAP server, see your LDAP server documentation.

-ldapattrusrunixuid=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for UNIX user ID.

Default: uidNumber

-ldapattrusrunixgid=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for UNIX primary group ID.

Default: gidNumber

-ldapattrusrunixhomedir=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for UNIX home directory.

Default: homeDirectory

-ldapattrusrunixshell=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for UNIX shell.

Default: shell

OpenLDAP External Authentication Connection Options (Servers)

The following options apply *only* to CA Harvest SCM servers.

Enter or modify these options in the server's HServer.arg file.

-ldapattrusrfullname=*attribute-name*

(Optional) Defines your LDAP server's user attribute name **for Full Name**.

Default: fullName

-ldapattrusrphone=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for Phone Number.

Default: telephoneNumber

-ldapattrusrphoneext=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for Phone Extension.

Default: telephoneExtension

-ldapattrusrfax=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for Fax Number.

Default: facsimileTelephoneNumber

-ldapattrusremail=*attribute-name*

(Optional) Defines your LDAP server's user attribute name for eMail.

Default: mail

Options for LDAP Attribute Names

Verify that the LDAP attribute names used by the product are the same as the attribute names used by your LDAP server. The product provides configuration parameters for the following LDAP attribute names:

- -ldapattrusrname
- -ldapattrusrfullname
- -ldapattrusrphone
- -ldapattrusrphoneext
- -ldapattrusrfax
- -ldapattrusremail
- -ldapattrusrunixuid
- -ldapattrusrunixgid
- -ldapattrusrunixhomedir
- -ldapattrusrunixshell

Example: Change the User Attribute Name

This example shows how to use the -ldapattrusrname attribute and specify a parameter to change the user's name to sAMAccountName.

```
-ldapattrusrname=sAMAccountName
```

Example: Prevent the Retrieval of a Users' Phone Number

If the LDAP server does not contain any values for a specific attribute, use an empty string as the attribute name. By using an empty string, the product does not try to retrieve the attribute and updates the value with blanks.

This example shows how to use the -ldapattrusrphone attribute and an empty string as the attribute name to prevent the retrieval of a users' phone number. As a result, the product populates the phone number value with blanks.

```
-ldapattrusrphone=""
```

SearchBase DN and Filtering Specifications

The LDAP authentication server API uses an OpenLDAP filter specification to help select the proper user account container from a set of all entries associated with the SearchBase DN (for example, the -ldapbasedn parameter). The filter specification uses a *pre-fix* operator format, rather than the more common *in-fix* operator format. For example, the default filter is the following expression:

```
(&(objectclass=person)(uid=<uid>))
```

This expression means: Search for directory entries that meet both of the following criteria:

- The user name attribute (uid) value equals the requested user identifier
- The object class attribute value equals “person”

In some cases, this level of filtering is not sufficient. For example, on certain directories that have *computer* containers as well as *user account* containers, the previous filter may return a computer container entry for the user rather than the user account container. In such cases, refine the filter to exclude computer node entries, for example:

```
(&(!(objectclass=computer))(&(objectclass=person)(uid=<uid>)))
```

This expression means: Search for directory entries that meet all of the following criteria:

- The user name attribute (uid) value equals the requested user identifier
- The object class attribute value equals “person”
- The object class attribute value does *not* equal “computer”

The time required for the search depends mostly on the selection of the Search Base DN. For example, the following expression searches an entire domain:

```
DC=mydomain,DC=com
```

When searching large domains, you can reduce the search time by limiting the search to a specific portion of the entire domain, for example:

```
OU=North America, DC=mydomain,DC=com
```

OpenLDAP Configuration Files Instead of TLS Startup Parameters

On UNIX, instead of specifying TLS parameters as command line or .arg file parameters, you can optionally define the TLS certificate and key specifications in an OpenLDAP configuration file located in $HOME/.ldaprc.

For example, you can add the following TLS parameters to the .ldaprc file:

TLS_CACERT "*filename*"

Defines the complete path name of the TLS trusted certificate file.

TLS_CERT "*filename*"

Defines the complete path name of the TLS client certificate file.

TLS_KEY "*filename*"

Defines the complete path name of the TLS private key associated with the client certificate file.

Securing Communication to the LDAP Server

You can use either Transport Layer Security (TLS) or Secure Sockets Layer (SSL) to encrypt the authentication and communication between the CA Harvest SCM server or remote agent and the LDAP v3 compliant directory server.

TLS is the default encryption mode and can be used for all directory servers that support the StartTLS extended operation of LDAP.

For LDAP directory servers that do not support the StartTLS extended operation, you can use SSL instead of TLS. For example, if your Microsoft Active Directory on Windows 2000 Server does not support StartTLS, you can use SSL instead of TLS.

Microsoft Active Directory servers must meet the following requirements to use SSL:

- Must support 128-bit encryption. See the Microsoft web site for details on obtaining and installing the high-encryption version of the latest service pack.
- Must have been issued a digital certificate.

Note: LDAP servers commonly use Port 389 as the default port for TLS security and Port 636 as the default port for SSL security.

The OpenLDAP Certificate and Key Files for TLS or SSL Encryption Mode

The CA Harvest SCM remote agents and servers that are using LDAP authentication with TLS or SSL security may require a trusted certificate file, a client certificate file, and a private key associated with the client certificate. If you are not certain whether these requirements apply to you, see your system administrator.

LDAP authentication supports the use of TLS and SSL by using TLS/SSL v3 and the OpenLDAP API to perform these major functions:

- Protect the user's logon credentials (user name and password) by using optional encryption security. By default, this encryption security is enabled. To enable or disable it, use the -ldapmode setting in each remote agent's or server's .arg file (HAgent.arg for the agent, HServer.arg for the server).
- Validate the user's logon credentials with an authentication mechanism that is always enabled.

For LDAP directory servers that do not support the StartTLS extended operation, you can use SSL instead of TLS. For example, if your Microsoft Active Directory on Windows 2000 Server does not support StartTLS, you can use SSL instead of TLS.

Note: For more information about how to create the certificate and key files, see the appropriate section for the authentication server you are using: Microsoft Active Directory, Novell eDirectory, or IBM Directory.

Microsoft Active Directory

For Microsoft Active Directory, you may need to create the trusted certificate file and to verify that SSL is enabled on the Active Directory Server. This section provides sample procedures for performing both tasks.

Create the Trusted Certificate File

Creating the trusted certificate file enhances security for authenticating and protecting users' logon credentials.

Follow these steps:

1. Select Start, Administrative Tools, Certificate Authority to open the CA Microsoft Management Console (MMC) GUI.
2. Right-click the CA computer and select Properties.
3. Click General, View Certificate.
4. Click the Details view and click Copy to File.

5. Use the Certificate Export Wizard to save the CA certificate in a file.

   Note: You can save the CA certificate in either DER Encoded Binary X-509 format or Base-64 Encoded X-509 format.

6. Use the OpenSSL utility to convert the CA certificate from DER to PEM format, because OpenLDAP requires PEM format. Use the following command as a model:

   ```
   openssl x509 -inform DER -in hldapads.cer -outform PEM -out hldapads.pem
   ```

   where hldapads.cer is the file generated in previous step.

   The resulting file, hldapads.pem, is used for the CA Harvest SCM remote agent TLS configuration.

Verify that SSL is Enabled

Verifying that SSL is enabled, also enhances security for authenticating and protecting users' logon credentials.

Follow these steps:

1. Verify that Windows Support Tools is installed on the Active Directory computer. The suptools.msi setup program is located in the \Support\Tools\ directory on your Windows installation media.
2. Open the Windows Support Tools and start the ldp tool.
3. From the ldp window, click Connection, Connect and supply the Active Directory domain server name and port number (636).

   If the connection succeeds, a window appears listing information related to the Active Directory SSL connection.

   If the connection fails, restart your system and repeat this procedure.

Novell eDirectory

For Novell eDirectory, you may need to create the trusted certificate file and to create the client certificate and private key files. Creating the trusted certificate file enhances security for authenticating and protecting users' logon credentials. Sample procedures for performing these tasks follow.

Follow these steps:

1. Start the ConsoleOne client for your Novell eDirectory server.
2. On the ConsoleOne console, right-click the IP AG object and select Properties from the shortcut menu.

   The Properties dialog appears.

3. Click the Certificates, Trusted Root Certificate configuration tab.
4. Click Export to export this certificate, but do *not* include the private key with the export.

5. Click DER format to export the key file.
6. Leave the Properties dialog open and start a command prompt.
7. Use the OpenSSL utility to convert the certificate from DER format to PEM format, because OpenLDAP requires PEM format. Use the following command as a model:

   ```
   openssl x509 -in cacertfile.der -inform DER -outform PEM -out cacertfile.pem
   ```

   where *cacertfile*.der is the file generated by ConsoleOne.

   The resulting file, *cacertfile*.pem, is used for the CA Harvest SCM remote agent TLS configuration.
8. Leave open the Properties dialog and the command prompt and complete the following steps to create the client certificate and private key files. When you create the client certificate and private key files, you enhance security for authenticating and protecting users' logon credentials.

Important! Before performing these steps, verify that you have completed the previous steps to create the trusted certificate file.

Follow these steps:

1. On ConsoleOne, select the user object (for example, CA Harvest SCM Admin) and open the Properties dialog.
2. On the Properties dialog, click the Security tab and select Certificates:
   a. Export this certificate and specify that the private key be included with the export.
   b. In the fields provided, enter and confirm the password for encrypting the private key.
   c. Record the password because you need it in a next step to separate the certificate from the private key.

      The client certificate is exported.

      Use the OpenSSL utility to separate the certificate from the private key and remove the password.
   d. Enter the following command to extract the certificate and remove the password:

      ```
      openssl pkcs12 -in certfile.pfx -clcerts -nodes -nokeys -out certfile.pem
      ```
   e. When prompted, enter the password used when exporting the pfx certificate created by ConsoleOne. In this example, the exported certificate file is *certfile.pfx*.

f. Enter the following command to extract the private key and remove the password:

```
openssl pkcs12 -in certfile.pfx -clcerts -out temp.pem -nodes
```

g. When prompted, enter the password used to export the pfx certificate:

```
openssl rsa -in temp.pem -out key.pem
```

The certfile.pem and key.pem files are used for the TLS configuration. The temp.pem file is an intermediate file and can be deleted.

IBM Directory

CA Harvest SCM works with IBM Tivoli Directory Server 5.2.

The GSKIT 7.0 toolkit bundled with the IBM Tivoli Directory Server must be installed to enable your system to generate certificate requests and self-signed root certificates. See the IBM Tivoli Directory Server documentation for the prerequisites and installation instructions for GSKIT.

Important! When you use the GSKIT 7.0 toolkit bundled with IBM Tivoli Directory Server 5.2, you may experience problems generating the certificate requests. To enable your system to generate certificate requests successfully, you may need to install the GSKIT 7.0.1.16 update patch.

For IBM Directory, you may need to configure your Java runtime environment and to create a self-signed root certificate. Sample procedures for performing both tasks follows.

Configuring the Java runtime environment is a prerequisite for creating the self-signed root certificate.

Follow these steps:

1. Verify that you have Java Runtime Environment installed.
2. Update the $JAVA_HOME/lib/security/java.security file to add IBM CMS security provider in the first position. For example:

   ```
   Security.provider.1=com.ibm.spi.IBMCMSProvider

   Security.provider.2=com.ibm.crypto.provider.IBMJCE
   ```

3. If applicable for your system, make any other required modifications to your Java runtime environment.

   Note: For detailed information, see the IBM Tivoli Directory Server documentation.

Creating the self-signed root certificate file enhances security for authenticating and protecting users' logon credentials.

Follow these steps:

1. Start the IBM Key Management utility.
2. Click Key Database File, New to create a server key database (.kdb file).

   The New dialog appears.
3. Verify that the key database type option is set to CMS.
4. Specify the filename and complete path for the key file and click OK to close the dialog.

   If you do not see the CMS option in the drop-down list, see the Important Note at the beginning of this section (IBM Directory) and verify that you have completed the steps in the previous section, Configure the Java Runtime Environment.
5. Click OK.

   The Password Prompt dialog appears.
6. Enter a password and set a password expiration time. For example, set an expiration time of 1000 days.
7. Click OK to complete the request.
8. In the main window, click Create, New Self-Signed Certificate. Provide the following information.

   Key label

   Defines a descriptive label for the certificate.

   Key version

   (Optional) Specifies the version of the key, typically X509 V3.

   Common name

   (Optional) Defines the common name of the LDAP server computer. This value is typically the computer's fully qualified domain name.

   Organization

   Defines your organization name.

   Validity

   (Optional) Defines the duration for which the certificate is valid.
9. Click OK.

   The request is created.

10. Select the new certificate's entry in the Personal Certificate list and click Extract Certificate.

    The Extract Certificate to a File dialog appears.

11. Select Base64-encoded ASCII data from the Data type list.

12. Enter a file name with an .arm extension and the complete path to which the root certificate is to be exported.

13. Click OK.

    The root certificate is exported.

    Note: The self-signed certificate, rootcert.arm, can be used by the LDAP clients, the product's broker and remote agent, to communicate with the IBM Tivoli Directory Server in the Transport Layer Security mode.

The CA Software Delivery Integration

If you install both CA Harvest SCM and CA Software Delivery in your environment, you can use special lifecycle processes in the product to deploy software from CA Software Delivery. This feature extends the lifecycle support provided by the product by adding the final phase of software development: delivering the updated software. You can optionally deploy to different targets at different times.

Using the predefined Deploy Release Model lifecycle, or a customized lifecycle similar to it, you can stage, create, and deploy CA Software Delivery packages for distribution to target computers and computer groups. You can use the product and CA Software Delivery together to deploy software on *any* platform that the CA Software Delivery agent supports.

Note: For details about these platforms, see your CA Software Delivery documentation.

CA Software Delivery Integration Configuration

You can configure CA Software Delivery Integration for CA Harvest SCM to use the product and CA Software Delivery together to deploy software in the following situations:

- You installed the product server *without* installing CA Software Delivery Integration, but now, for the first time, you want to configure the product to use CA Software Delivery Integration, but you do not want to re-install the server.
- You installed the server and you *did* install and configure CA Software Delivery Integration, but now you want to *reconfigure* your CA Software Delivery Integration settings.
- You installed the server without CA Software Delivery Integration originally and you later configured the product to use CA Software Delivery Integration, but now you want to *reconfigure* your CA Software Delivery Integration settings.

Reconfiguring your CA Software Delivery Integration settings may be required by the security software or company policy in your environment. For example, security software may require you to update user names and passwords regularly.

Note: For more information about installing CA Software Delivery Integration when you install the server, see CA Software Delivery Integration (see page 20).

How To Enable the CA Software Delivery Integration

To enable or reconfigure CA Software Delivery Integration, set the following parameters to configure the communication and the synchronization of data between CA Harvest SCM and CA Software Delivery.

1. Set the URL for the CA Software Delivery server.
2. Set the logon credentials (user name and password) for accessing the CA Software Delivery server from the product.
3. Set the logon credentials for accessing the product remote agent running on the CA Software Delivery server.
4. Set the port number for accessing the product remote agent running on the CA Software Delivery server.
5. Set the synchronization interval between the product and the CA Software Delivery server.
6. Import the Deploy Release Model lifecycle template by running himpenv, the product import command-line utility.

Implement CA Software Delivery Integration Using Configuration Utilities

You configure CA Harvest SCM for ITCM Software Delivery integration using provided utility programs.

On Windows, use the hsdsetup utility program, which displays a dialog that accepts all the required parameters and stores them in the appropriate configuration files.

On UNIX and Linux, run the install.sh script using the –configusd option.

Example: Run install.sh

```
./install.sh -configusd
```

You manually configure the ITCM Software Delivery Integration parameters.

Set the URL for the CA Software Delivery Server

Setting the URL for the CA Software Delivery server is the first major step in establishing communication between CA Harvest SCM and CA Software Delivery, a prerequisite step for using the CA Software Delivery Integration to deploy software.

Follow these steps:

1. Shut down the product server used to connect to the CA Software Delivery server.
2. Shut down the product broker for that product server.
3. Shut down all other product servers and Web Interface servers (if any) connected to that broker.
4. Edit the product server's \CA_SCM_HOME\HServer.arg file and set the following parameter. If necessary, enter the entire parameter=value statement.

   usdserver=*URL*

   For a CA Software Delivery server running on *Windows*, use the following format for the URL:

   `http://server hostname or network address/UDSM_R11_WebService/mod_gsoap.dll`

   For a CA Software Delivery server running on *Linux*, use the following format for the URL:

   `http://server hostname or network address/UDSM_R11_WebService`

   Default: None

   Limits: 255 characters

Examples:

To specify the URL for a CA Software Delivery server on Windows, using the server's name (usdsrv01), enter the following parameter:

```
usdserver=http://usdsrv01/UDSM_R11_WebService/mod_gsoap.dll
```

To specify the URL for a Linux server whose network address is 138.42.44.57, enter the following parameter:

```
usdserver=http://138.42.44.57/UDSM_R11_WebService
```

Note: For more information about this URL format, see the information about the login web service in the *Unicenter Desktop and Server Management (DSM)* documentation.

5. Restart the product broker and its connected product servers and Web Interface servers.

Set the Logon Credentials for Accessing the CA Software Delivery Server

If you install the CA Software Delivery Integration when you install the CA Harvest SCM server, the product automatically records the user name and password (if specified) that you enter for accessing the CA Software Delivery server from the product. The product stores these logon credentials in the husdsr.dfo file and encrypts this file, using the svrenc utility.

The husdsr.dfo file is a hidden file in the CA_SCM_HOME directory on the product server. After the product server installation, to create or update the encrypted user name, password, or both for accessing the CA Software Delivery server from the product, you must use the svrenc utility.

Important! Do *not* change the file name husdsr.dfo and do *not* change its location, the CA_SCM_HOME directory.

On Windows, enter the following command:

```
svrenc -f husdsr.dfo -usr username -pw password
```

On UNIX and Linux, enter the following command:

```
./svrenc -f husdsr.dfo -usr username -pw password
```

husdsr.dfo

Defines the file name that stores the encrypted user name and password.

username

Defines the user name for the product to use to log on to the CA Software Delivery server.

Note: The logon credentials for the CA Software Delivery server and the product remote agent on the CA Software Delivery server are *not* validated when you create or update the husdsr.dfo file. Instead, they are validated when you attempt to invoke one or more "USD" processes from the product or to synchronize CA Software Delivery and the product databases.

For a CA Software Delivery server running on *Windows*, use the following format for the user name:

winnt://*login domain or host name*/*username*

For a CA Software Delivery server running on Linux, use the following format for the user name:

unixl://*login domain or host name*/*username*

Default: None

Limits: 255 characters

password

(Optional) Defines the password for this user. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

Example 1-Windows

For a CA Software Delivery server on Windows, to create or update the husdsr.dfo file in the CA_SCM_HOME directory to store the user name *usdadmin*, using the name of the user's domain *domain01* and password *cascm,* enter the following command:

```
svrenc -f husdsr.dfo -usr winnt://domain01/usdadmin -pw cascm
```

Example 2-UNIX or Linux

For a CA Software Delivery server on Linux, to create or update the husdsr.dfo file in the CA_SCM_HOME directory to store the user name *usdadmin*, using the host name *usdsrv01* of the CA Software Delivery server and password *cascm*, enter the following command:

```
./svrenc -f husdsr.dfo -usr unixl://usdsrv01/usdadmin -pw cascm
```

Note: For more instructions about how to use svrenc, see the *Command Line Reference Guide* or display the command line help by entering one of the following commands from the CA_SCM_HOME directory. The first command is for Windows, the second for UNIX and Linux.

svrenc -h

```
./svrenc -h
```

Set the Logon Credentials and Port Number for Accessing the Remote Agent

If you install the CA Software Delivery Integration when you install the CA Harvest SCM server, the product automatically records the user name and password (if specified) that you enter for accessing the product remote agent on the CA Software Delivery server. The product stores these logon credentials in the husdra.dfo file and encrypts this file, using the svrenc utility.

The husdra.dfo file is a hidden file in the CA_SCM_HOME directory on the product server. After the product server installation, to create or update the encrypted user name, password, or both for accessing the product remote agent on the CA Software Delivery server from the product, you must use the svrenc utility.

Important! Do *not* change the file name husdra.dfo and do *not* change its location, the CA_SCM_HOME directory.

On Windows, enter the following command:

```
svrenc -f husdra.dfo -usr username -pw password
```

On UNIX and Linux, enter the following command:

```
./svrenc -f husdra.dfo -usr username -pw password
```

husdra.dfo

Defines the file name that stores the encrypted user name and password.

USD CA Harvest SCM Agent User Name

Defines the user name for the product to use to access the product remote agent running on the CA Software Delivery server.

Note: CA Software Delivery is not required to be installed before the product server is installed. Therefore, the logon credentials for the CA Software Delivery server and the product remote agent on the CA Software Delivery server are not validated at installation time. Instead, they are validated when you attempt to invoke one or more "CA Software Delivery" processes from the product or synchronize CA Software Delivery and the product databases.

Default: None

Limits: 255 characters

USD CA Harvest SCM Agent Password

(Optional) Defines the password of the user specified in the USD CA Harvest SCM Agent User Name field. Do *not* enter spaces. If you do not specify a password, an empty password is used.

Default: None

Limits: 255 characters

Example 1-Windows

On Windows, to create or update the husdra.dfo file in the CA_SCM_HOME directory to store the user name *cascm* and password *cascm*, enter the following command:

```
svrenc -f husdra.dfo -usr cascm -pw cascm
```

Example 2-UNIX or Linux

On UNIX or Linux, to create or update the husdra.dfo file in the CA_SCM_HOME directory to store the user name *cascm* and password *cascm*, enter the following command:

```
./svrenc -f husdra.dfo -usr cascm -pw cascm
```

Edit the product server \CA_SCM_HOME\HServer.arg file and set the following parameter.

```
-scmusdagentport=port
```

Default: None

Limits: 0-9999

Example: Specify the Agent Port Number

To specify the agent port number of 1234, enter the following parameter:

```
-scmusdagentport=1234
```

Set the Synchronization Interval

Setting the synchronization interval between CA Harvest SCM and the CA Software Delivery server is a critical task for maintaining CA Software Delivery information, including status, at the product server.

Important! If you do not set this parameter, *no* synchronization occurs between the product and the CA Software Delivery server.

Follow these steps:

1. Shut down the product server used to connect to the CA Software Delivery server.
2. Shut down the product broker for that product server.
3. Shut down all other product servers and Web Interface servers (if any) connected to that broker.
4. Edit the product broker's \CA_SCM_HOME\HBroker.arg file and set the following parameter. If necessary, enter the entire parameter=value statement.

   usdsynchinterval[*_version*]= *minutes*

   Defines the interval (in minutes) at which the product synchronizes certain tables in the product database with those in the CA Software Delivery database. The product checks the contents of these tables in the CA Software Delivery database and, if necessary, updates the product tables to match the CA Software Delivery tables.

   At this synchronization interval, the product also queries for the status of any outstanding deployment jobs it has scheduled. For those jobs whose status has changed, the HARUSDPLATFORMINFO and HARUSDHISTORY tables are updated.

   _version

   (Optional) Specifies the broker version. When no version is specified, the version name “Default” is used.

   minutes

   Defines the synchronization interval in minutes.

   Default: None. If you do not specify a value, *no* synchronization will occur.

   Minimum: 15

   Limits: 6 digits
5. Restart the product broker and its connected product servers and Web Interface servers.

Import the Deploy Release Model Life Cycle Template

Run himpenv, the CA Harvest SCM import command line utility, to import the Deploy Release Model lifecycle template.

Note: The Deploy Release Model lifecycle template is required to use the CA Software Delivery Integration; this lifecycle template is not installed automatically, even if you install the CA Software Delivery Integration when you install the product server.

Use the following syntax as a model:

```
himpenv -b broker f DeployReleaseModel.har usr scm_username pw scm password
```

-b *broker*

Specifies the name of your product broker.

f DeployReleaseModel.har

Specifies the complete path name of the Deploy Release Model lifecycle template. This file resides in the CA_SCM_HOME directory.

usr *scm_username*

Specifies the user name of a product user with administrator rights.

pw *scm password*

Specifies the password for this administrative user.

Note: For more information about using the himpenv command, see the *Command Line Reference Guide*.

Disable the CA Software Delivery Integration

To disable the CA Software Delivery Integration, delete the related parameters from the HBroker.arg and HServer.arg files. If you decide to no longer use CA Software Delivery Integration, deleting these parameters helps CA Harvest SCM run more efficiently.

Follow these steps:

1. Shut down the server used to connect to the CA Software Delivery server.
2. Shut down the broker for that server.
3. Shut down all other servers and Web Interface servers (if any) connected to that broker.
4. Delete the usdserver=URL and scmusdagentport= value parameters from the server's \CA_SCM_HOME\HServer.arg file, and save the file. These parameters define the URL for the CA Software Delivery server and CA Harvest SCM agent port on CA Software Delivery server computer.

5. Delete the usdsynchinterval[_version]= minutes parameter from the broker's \CA_SCM_HOME\HBroker.arg file, and save the file. This parameter defines the synchronization interval between the product and the CA Software Delivery Server.
6. Restart the broker and its connected servers and Web Interface servers.

Connect Method Options for Direct Connection

Two connect method options, -cm and -cma, can be used to specify the connect method for the creation of a direct connection. The option -cm can be specified for CA Harvest SCM server and client in the HServer.arg and HClient.arg files, respectively. The option -cma can be specified only for the product client in the HClient.arg file. Both options take one of the two values, i (initiate connection) and a (accept connection). In the product, the direct connection can be created for the product server and the product client.

The product client uses -cm to specify the connect method for the connection to the product server. The connect method priority is used to determine the final connect method, because both ends of a connection can have the connect method specified. The connect method priorities for the product server and client are in the following order:

- Client (highest priority)
- Server (lowest priority)

Direct Connection Methods

The final connect method is determined based on the connect method option specified for each end of a connection and the connect method priority. The final connect method is used to create the direct connection.

1. When no connect method option is specified for both ends of a connection, the default action is taken as follows:

   For a connection between the CA Harvest SCM server and the client, the hclient initiates the connection and the hserver accepts the connection.

2. When both ends of a connection have the connect method specified, the final connect method is determined based on the following connect method priorities:

   For a connection between the server and the client, the hclient connect method overrides the hserver connect method.

3. When only one end of the connection has the connect method specified, the specified connect method is used as the final connect method.

The Agent Trace Facility

To turn on a certain trace mode for the CA Harvest SCM agent, specify the option -trlvl=*int* in the product agent command line or in the HAgent.arg file.

When the trace option is on and the option -verbose is not specified, trace messages are logged into the agent log file. If the product agent is running in verbose mode (with -verbose), trace messages display in the standard output.

Trace messages contain five fields: thread ID, time stamp, trace mode, function name, and additional information.

Thread id | *Time stamp* | **LVLx:** *Function Name - Additional Information*

The following example shows a mode 1 trace with thread ID 13, time stamp 08:23:29, function name ConnectionMsgThreadMain, and no additional information.

```
13 | 08:23:29 | LVL1: in ConnectionMsgThreadMain
```

The following example shows a mode 2 trace with thread ID 420, time stamp 17:09:57, function name CptHAgentC::Fremove, and information that remove file c:\cko1\f1.sh has failed.

```
420 | 17:09:57 | LVL2: CPtHAgntC::FRemove failed to remove <C:\cko1\f1.sh>
```

The following example shows a level 3 trace with thread ID 368, time stamp 16:45:31, function name cbExecuteA, following the successful invocation of "hexecp -b xxx -m xxx -usr xxx -pw xxx -syn -prg c:\1651\echo.bat":

```
368 | 16:45:31 | LVL3: cbExecuteA bSynch is true 41| File<c:\1651\echo.bat> CmdArg<>
Stdin<> CB<4211056> Fm</pt_HClient://ni-ya01h511b2/62> |0

368 | 16:45:35 | LVL3: cbExecuteA done 41| File<c:\1651\echo.bat> |0
```

In addition to four modes of trace, an error message is always logged regardless of the trace mode specified. The error message indicates that an agent encountered an error but it is not able to reply to the client. The error message has the following format:

Thread id | *Time stamp* | **ERR:** *Function Name - Additional Information*

Example:

```
127 | 08:39:08 | ERR: cbLogIn() -ServerCrypt.CreateKeyAgreementPair() failed
```

The Server Logging Option

The -logging=*level* option in the CA Harvest SCM server command line or in the HServer.arg file let you use logging and set a logging level for the product server.

To use logging and set a logging level for the product server, specify the option *-logging=level* in the product server command line or in the HServer.arg file. Each log level displays the information of all levels below it. The levels are cumulative not mutually exclusive. In the -logging=*level* operand, specify a level from 1-4, as follows:

1. Displays ODBC errors and any error messages written by the relational database.
2. Displays the product Transaction name; the date and time it started; the date and time it ended; the name of the server and process ID that processed the transaction; and the duration of the transaction in milliseconds.
3. Displays the executed SQL statement, when it started, when it ended, the execution time in milliseconds, the time in milliseconds since the last completed SQL.
4. Displays commit and rollback statements.

When the product server is running with the logging option on and verbose mode off (the option -verbose is not specified), logging messages are recorded in the server log file. If verbose mode is on, log messages display in the standard output.

The following example shows a level 1 logging for an administrator attempting to define a user having a name that exists:

```
HServer | 20050519 14:49:00 | ---- Started ----

[CAI/PT][ODBC Oracle 8 driver][Oracle]ORA-00001: unique constraint
(CASCM.HARUSER_IND) violated

SQLSTATE=23000
```

The following example shows a level 2 logging for project rename in the Administrator application:

```
2005-05-19 21:41:02 | Start UPDATE_ENVIRONMENT /pt_HClient://computer1/5576

2005-05-19 21:41:02 | End UPDATE_ENVIRONMENT   DURATION (msec): 0
```

The following example shows a level 3 logging for project rename in the Administrator application:

```
2005-05-19 21:32:49 | Start UPDATE_ENVIRONMENT /pt_HClient://computer1/6184

-- START: 2005-05-19 21:32:49  SID: 25 (SINCE LAST: 0)

UPDATE  HARENVIRONMENT SET ENVIRONMENTNAME = 'new project 2', ENVISACTIVE = 'Y', NOTE
= '', MODIFIEDTIME = {TS '2005/05/19 21:32:49'}, MODIFIERID = 1 WHERE ENVOBJID = 94

-- END:  2005-05-19 21:32:49    **** SQL Milliseconds: 16

-- COMMIT

2005-05-19 21:32:49 | End UPDATE_ENVIRONMENT   DURATION (msec): 16
```

The following example shows a level 4 logging for project rename in the Administrator application:

```
2005-05-19 21:30:18 | Start UPDATE_ENVIRONMENT   /pt_HClient://computer1/3972

-- START: 2005-05-19 21:30:18  SID: 17 (SINCE LAST: 31)

UPDATE  HARENVIRONMENT SET ENVIRONMENTNAME = 'new project', ENVISACTIVE = 'Y', NOTE
= '', MODIFIEDTIME = {TS '2005/05/19 21:30:18'}, MODIFIERID = 1 WHERE ENVOBJID = 94

-- END:  2005-05-19 21:30:18    **** SQL Milliseconds: 31

-- COMMIT

-- ROLLBACK (usually part of Database connection cleanup)

2005-05-19 21:30:18 | End UPDATE_ENVIRONMENT   DURATION (msec): 31
```

The following example shows a level 5 logging for project rename in the Administrator application. In level 5, you can see the container content.

```
-- START: 2006-01-12 12:10:22  DB Connect#: 0  SINCE LAST SQL (msec): 0

   select BUILDNUMBER

   from tau_mdb

-- RETURN CODE: DB_SUCCESS

-- END:  2006-01-12 12:10:22    **** SQL Milliseconds: 31

-- SQLFreeStmt(SQL_CLOSE)

-- CONTAINER CONTENT

      BUILDNUMBER

        0       30
```

Broker Setup for Multiple Server Instances

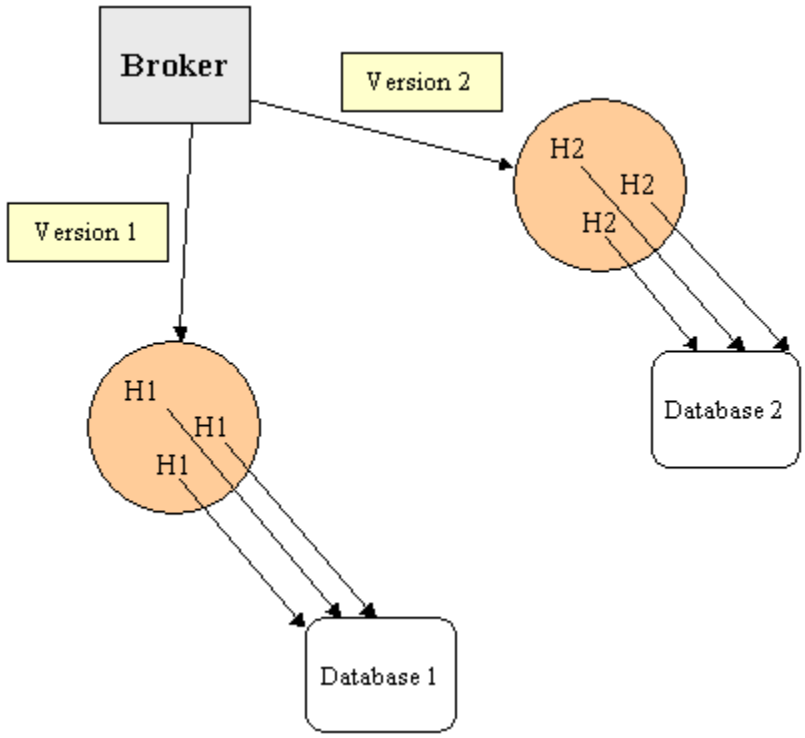
You can set up the broker daemon to support multiple instances of CA Harvest SCM. Conceptually, you use the same broker to manage different sets of server processes, with each set connecting to its own distinct database instance. These sets are referred to as different versions.

Note: Load-balancing cannot be used with multiple versions of server processes.

A brief review of this setup follows:

- One broker daemon starts and manages different pools of server processes.
- Each pool of server processes has its own HOMESERVER directory that stores the distinct database information (datasource, username, and password) in the HServer.arg file.
- When a client connects, it must specify the broker version in the Broker dialog box, and then the broker daemon connects that client to only the server processes in the appropriate version pool.

The following illustration shows broker daemon versions communicating with pools of server processes, and the server processes communicating with a specific database. Each version corresponds to a connection from the broker to a particular pool of server processes (H1 or H2), and then to the database.



Configure the Servers for Database Instances (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can configure CA Harvest SCM servers to connect with distinct database instances. The server must be installed prior to performing this step.

Follow these steps:

1. For each database instance, create a CAI/PT ODBC datasource connection to the database instance.
2. In the $ODBC_HOME/bin directory, run the configdsn utility to define an ODBC datasource name (DSN).
3. Select option 1 to add a new datasource.
4. Enter a name to identify the datasource. For example, enter *CA Harvest SCM* to indicate a datasource to be used with the product.

   Note: Make note of the datasource name, as it will be used later in the installation. In addition, datasource names for CAI/PT ODBC cannot include an underscore character '_'.
5. Select an existing driver definition by number. Enter the number corresponding to the appropriate CAI/PT DBMS driver.
6. Follow the DMBS-specific questions:

   ORACLE

   a. (Optional) Enter a description).

   b. Enter the Oracle Net*8 global alias name that identifies the database instance.

   c. Enter the following semicolon separated arguments to properly configure the SQLDriverConnect string at the SQLDriverConnect format prompt:

      ```
      EnableScrollableCursors=1;ApplicationUsingThreads=1
      ```

d. (Optional) Enter the server type.

"ServerType" refers to the class of Net server accessing the database server installation. If the database server installation is being accessed through an Enterprise Access or EDBC server, specify the gateway server. Valid definitions are:

- **Advantage EDBC**-DCOM (Datacom), IDMS, DB2, IMS, VSAM
- **Advantage Enterprise Access**-RDB, RMS, ORACLE, INFORMIX, SYBASE, MSSQL and DB2UDB

e. Enter the "BlankDate" Value (optional).

a. Valid settings are Y (default) or N. N causes the ODBC driver to return the date value of 9999-12-31 23:59:59 for empty date values. Most applications prefer to receive a more meaningful NULL value when displaying an empty string date.

b. Enter an optional semicolon-separated list of other records (optional).

The product does not require a list of additional arguments to this prompt.

7. Select option number 6 to exit the program.
8. Create a version directory in the $CA_SCM_HOME for each server version. The default server version uses the $CA_SCM_HOME location. A server version directory is needed only for a nondefault server.

   For example, suppose you are using the default server and two server versions named version1 and version2. Create two version directories in the $CA_SCM_HOME location: $CA_SCM_HOME/version1 and $CA_SCM_HOME/version2.
9. In each server version directory, create a log directory. For this example, create $CA_SCM_HOME/version1/log and $CA_SCM_HOME/version2/log.
10. Copy the HServer.arg file in the $CA_SCM_HOME directory into each server version directory.
11. Edit each HServer.arg file. Change the datasource name to a datasource created in Step 1. The datasource name follows the -datasource= flag in the HServer.arg file.
12. Modify the HBroker.arg file found in the $CA_SCM_HOME directory to specify the startup parameters for the server versions. The parameters are:

    -dirserver[_version]=dir

    Specifies the directory containing the server process. Replace *dir* with the full path to the $CA_SCM_HOME/bin directory.

    -minserver[_<version>]=<int>

    Specifies the minimum number of server processes to be started.

-maxserver[_<version>]=<int>

Specifies the maximum number of server processes to be started.

-homeserver[_<version>]=<home >

Specifies the directory containing the server process argument file, HServer.arg. The default is the CA_SCM_HOME location.

-queuesize[_<version>]=<int>

Specifies the number of requests allowed in queue before the broker starts a temporary server.

The following is a sample HBroker.arg file based on the previous example.

```
//Default Version: Broker
-dirserver=/home/ca scm-d/bin
-minserver=5
-maxserver=50
-queuesize=2


// Version: Broker/version1
-dirserver_version1=/home/ca scm-d/bin
-minserver_version1=3
-maxserver_version1=10
-queuesize_version1=10
-homeserver_version1=/home/ca scm-d/version1
// Version: Broker/version2
-dirserver_version2=/home/ca scm-d/bin
-minserver_version2=3
-maxserver_version2=10
-queuesize_version2=10
-homeserver_version2=/home/ca scm-d/version2
```

When the broker is started, five default servers, three servers of version1, and three servers of version2 are also started. When no version tag is appended to a parameter name, the default value is used. In the homeserver directory, an HServer.arg must exist that specifies the configuration for that pool of server processes such as their distinct (nondefault) database information. The default location of homeserver is the $CA_SCM_HOME directory.

13. Store and encrypt the database username and password using the svrenc utility in the $CA_SCM_HOME/bin directory. For instructions to use svrenc, enter the following command from the $CA_SCM_HOME/bin directory:

    ```
    ./svrenc -h
    ```

    This utility creates a hidden file that stores the encrypted database username and password for the product server. For example, to create an encrypted file for the server version1 using a database user named admin, execute the following command:

    ```
    ./svrenc -s -usr admin -pw admin -dir /home/ca SCM/version1
    ```

    Note: This utility has already been run for the "default" server during installation.

14. Start the broker from the $CA_SCM_HOME/bin directory by entering the following command:

    ```
    ./bkrd
    ```

    To connect to a particular version of the product, enter *brokercomputer/versionname* into the Broker field for the GUI or command line argument. To connect to server version1 using broker computer cascmcm, specify the broker name as cascmcm/version1.

More information:

Install the Server (Typical Installation) (see page 93)

Configure the Servers for Database Instances (Windows)

You can configure CA Harvest SCM servers to connect with distinct database instances. The server must be installed prior to performing this step.

Follow these steps:

1. Create an ODBC datasource name with a corresponding connection for each server version (except for the default product server).
   a. In the Windows Control Panel, open ODBC Data Sources.
   b. Select the System DSN tab.

2. Add a new system data source by selecting the applicable DBMS driver.
   - For Oracle, use the Oracle version of the driver.
   - For SQL Server, use the SQL Server driver.
3. (Oracle only) In the ODBC Oracle Driver Setup dialog, assign the new datasource a name. The description field of the datasource is optional.
   a. In the Server Name field, enter the Oracle Net*8 global alias name that identifies the database instance.
   b. Select the Advanced tab and place a check mark in the boxes Enable Scrollable Cursors and Application Using Threads.
   c. Select OK to complete the ODBC driver setup.

      The datasource is named.
4. (SQL Server only) In the SQL Server Driver Setup dialog, assign the new datasource a name. The description field of the datasource is optional.
   a. From the server drop-down list, select the server to which you want to connect.
   b. Select the authentication mode you want to use: Windows authentication or SQL Server authentication.
   c. (Optional) Click Client Configuration to change the network library you are using to communicate with SQL Server.

      Note: For more information about how to change the network library, see your SQL Server documentation.
   d. Click the check box named “Connect to SQL Server to obtain default settings for the additional configuration options.”
   e. In the user name and password fields under this check box, enter the user name and password, if you are using SQL Server authentication mode. If you are using Windows authentication mode, you are not required to complete these fields.
   f. Click the check box named “Change the default database to:”
   g. In the database drop-down list, select the database to which you want to connect.
   h. Accept the default settings for all other options.

      The datasource is named.

5. Create a version directory in the %CA_SCM_HOME% for each server version. The default server version uses the %CA_SCM_HOME% location. A server version directory is needed only for a non-default server.

   For example, suppose you are using the default server and two server versions named version1 and version2. Create two version directories in the %CA_SCM_HOME% location: %CA_SCM_HOME%\version1 and %CA_SCM_HOME%\version2.

6. In each server version directory, create a log directory; for this example, create directories named %CA_SCM_HOME%\version1\log and %CA_SCM_HOME%\version2\log.

7. Copy the HServer.arg file in the %CA_SCM_HOME% directory into each server version directory.

8. Edit each HServer.arg file. Change the datasource name to a datasource created in Step 1. The datasource name follows the -datasource= flag in the HServer.arg file.

9. Modify the hbroker.arg file found in the %CA_SCM_HOME% directory to specify the startup parameters for the server versions. The parameters include the following:

   -minserver[_<version>]=<int>

   Specifies the minimum number of server processes to be started.

   -maxserver[_<version>]=<int>

   Specifies the maximum number of server processes to be started.

   -homeserver[_<version>]=<home >

   Specifies the directory containing the server process argument file, HServer.arg. The default is the %CA_SCM_HOME% location.

   -queuesize[_<version>]=<int>

   Specifies the number of requests allowed in queue before the broker starts a temporary server.

The following is a sample hbroker.arg file based on the previous example.

```
//Default Version: Broker
-minserver=5
-maxserver=50
-queuesize=2


// Version: Broker/version1
-minserver_version1=3
-maxserver_version1=10
-queuesize_version1=10
-homeserver_version1=C:\Program Files\CA\SCM\version1


// Version: Broker/version2
-minserver_version2=3
-maxserver_version2=10
-queuesize_version2=10
-homeserver_version2=C:\Program Files\CA\SCM\version2
```

When the broker is started, five default servers, three servers of version1, and three servers of version2 are also started. When no version tag is appended to a parameter name, the default value is used. In the homeserver directory, an HServer.arg must exist that specifies the configuration for that pool of server processes such as their distinct (nondefault) database information. The default location of homeserver is the %CA_SCM_HOME% directory.

10. Store and encrypt the database username and password using the svrenc.exe utility in the %CA_SCM_HOME% directory. For instructions to use svrenc.exe, enter the following command at the command line:

    ```
    svrenc -h
    ```

This utility creates a hidden file that stores the encrypted database username and password for the product server. For example, to create an encrypted file for server version1 using a database user named admin, execute the following command:

Important! Be sure to use the double quotes (") if the value contains spaces.

```
C:\Program Files\CA\SCM\version1>svrenc -usr DBusername -pw DBpassword -dir
"C:\Program Files\CA\SCM\version1"
```

```
C:\Program Files\CA\SCM\version2>svrenc -usr DBusername -pw DBpassword -dir
"C:\Program Files\CA\SCM\version2"
```

After each command, a message confirms that the database information is successfully set.

Note: This utility has already been run for the "default" server during installation.

11. Start the broker from the product's program group by double-clicking bkrd.exe.

Note: To connect to a particular version of the product, enter *brokercomputer/versionname* into the Broker field for the GUI or command line argument. To connect to server version version1 using broker computer cascmcm, specify the broker name as cascmcm/version1.

More information:

Install the Server (Typical Installation) (see page 30)

Shut Down the Broker

The broker can be shut down from the local computer only; the broker cannot be shut down from a remote computer.

To execute a shutdown from the local computer (the computer on which the broker is running), enter the following command from the DOS or UNIX command prompt:

```
bkrd -shutdown
```

To shut down the broker and all CA Harvest SCM agents visible in the RTserver network, execute the following command from the DOS or UNIX command prompt:

```
bkrd -shutdown=all
```

Performance Tuning

To improve the performance of the product with Oracle or SQL Server, try the following:

- If you are denying access to many groups and many items, such as hundreds of items on an item path, delete the path from the baseline.

In addition, if you are using Oracle, try the following:

- The product installation includes three tablespaces for three different types of data: Index, Version data (BLOB), Logical data (metadata).

  You can take advantage of this outlay to improve performance by separating the media for each tablespace to increase the parallel access. If the media is a high-speed disk like RAID, performance is even better.

- The product servers running in the same system as the DBMS instance will offer best performance. If they are running on the same system, verify that the ODBC data source and DBMS instance are configured to allow servers to connect to the database locally. Doing so bypasses the network communication software, therefore increasing overall processing performance.

Event Audits

You can configure the CA Harvest SCM broker and server to record the success or the failure of product events. The following product events are available for auditing:

- Create a Project
- Update a Project
- Secure a Project
- Delete a Project
- Set Access or Duplicate Access to a Project Create State
- Update State
- Delete State
- Set Access or Duplicate Access to a State
- Create a State Process
- Update a State Process

- Secure a State Process
- Delete a State Process
- Set Access or Duplicate Access to a State Process
- Configure a Baseline View
- Secure a Baseline View
- Create a Working View
- Update a Working View
- Delete a Working View
- Secure a Working View
- Create a Linked Process
- Update a Linked Process
- Delete a Linked Process
- User Login
- Delete a Snapshot View

The audit log is created for successful completion of create, update, secure, configure, and delete actions. User login action, however, is logged only when the login fails.

How to Disable Event Auditing

Auditing is enabled by default on installation or upgrade to the CA Harvest SCM installation. The database table, HARAUDITLOGVIEW stores the audit events information. You can disable auditing by setting the option -auditenabled to 0 in the server configuration file (hserver.arg):

```
-auditenabled=0
```

Note: For more information about how the administrator uses the HARAUDITLOGVIEW, see the *Administrator Guide*.

Enterprise Communicator (PEC)

Enterprise Communicator (referred to as PEC) is a common communication product that CA Harvest SCM uses and requires. All PEC components are installed when you install the product. However, PEC can also run stand-alone as a separate component. Other products using PEC can assume and rely on all features of PEC being present when the RTHOME environment variable is set.

The RTserver

RTserver is a publish-subscribe message router that uses connections to make large scale distributed inter-process communication (IPC) easier. RTserver routes messages between the CA Harvest SCM server and the product client.

The RTclient

An RTclient is a process that is connected to an RTserver: the CA Harvest SCM client (GUI or command line) and the product broker process. Each RTclient has one IPC connection to one RTserver. An RTclient cannot be connected to more than one RTserver at a time.

The product broker has the capability to auto-start an RTserver process if one is not already running. Other RTclients and client GUI/command line, for example, automatically connect to the RTserver running on the product broker node.

RTserver Options

RTserver allows the setting of options at startup time. These options are defined in a startup command file named rtserver.cm. This file is located in the RTHOME/standard directory. The option values that have been specified in the RTserver command file are set each time RTserver is started.

Options are set with the setopt command. The general format is

`setopt` *`option values`*

server_names

Specifies a list of logical connection names used to find other RTserver processes. Each logical connection name has the form protocol:node:address, which can be shortened to protocol:node, or node.

Example option setting:

```
setopt server_names CA SCM
```

CA Harvest SCM specifies a node or computer name.

conn_names

Specifies a list of logical connection names used by RTclient processes (CA Harvest SCM client) to find this RTserver. Each logical connection name has the form protocol:node:address, which can be shortened to protocol:node, or node.

Example option setting:

```
setopt conn_names CA SCM
```

CA Harvest SCM specifies a node or computer name.

Logical Connection Names

A logical connection name has the form protocol:node:address. A server uses a logical connection name to create a server connection, and a client process uses the same logical connection name to create a client connection to the server process. For the client process to find the server process, the logical connection name used by the client must *exactly* match the logical connection name used by the server (for example, the name tcp:sparc:1234 does not match the name tcp:risc:1234).

Protocol

The protocol part of the connection name refers to an IPC protocol type. The valid values for protocol are:

`UNIX: local, tcp.`

`Windows: tcp.`

Node

The *node* part of the connection name refers to a computer node name or IP address. The special value _node can be used for node to indicate the name of the current node. This is the default if not specified.

Address

The address part of the connection name refers to a protocol-specific IPC location, such as a TCP port number.

Name-to-IP Address Resolution Requirements

Direct connection from client to server, client to a remote agent, a remote agent to a server, and server to remote agent is based on the following requirements:

Connection Type	Example	Requirement
Client to server	Check out to a local file system	Client host must be able to resolve the server's node name to an IP address.
Client to remote agent	Log in to a remote agent	Client host must be able to resolve the remote agent's node name to an IP address.
Remote agent to server	Check out to a remote agent	Remote agent host must be able to resolve the server's node name to an IP address.

Connection Type	Example	Requirement
Server to remote agent	Check in, Check out	Server must be able to resolve the agent host's name to an IP address.

Note: By default, CA Harvest SCM uses the computer's host name as the node name.

The RT_FORCE_NODE_NAME Environment Variable

Enterprise Communicator uses the computer node name returned by the hostname command (Windows) or uname command (UNIX). In some cases, it is necessary to override the default node name using an alias or virtual machine name.

The environment variable, RT_FORCE_NODE_NAME, forces Enterprise Communicator to use the specified node name. This variable is set in the user environment responsible for the startup of the RTserver and CA Harvest SCM broker. The product broker and product client use the assigned node name defined by the RT_FORCE_NODE_NAME variable.

- On Windows platforms, define the system variable, RT_FORCE_NODE_NAME, and assign it a node name. The value or node name assigned overrides the default node name used by Enterprise Communicator.
- On UNIX platforms, define an environment variable, $RT_FORCE_NODE_NAME, in the user shell. Set the value of this variable to override the default node name used by Enterprise Communicator.

You must restart RTserver, the product broker and the product clients for the change to take effect.

The Fully Qualified Domain Name as Node Name

CA Harvest SCM requires Name to IP Address resolution when establishing a connection from HClient to HServer, HClient to HAgent, and HAgent to HServer. By default, the product uses hostname (shortname, nodename) resolution when creating these connections.

In an environment where hostname resolution is not supported across the entire network and Fully Qualified Domain Name (FQDN) is the only name resolution infrastructure provided, it is necessary to enable the use of FQDN in the product components that require it.

On every computer running a remote agent or server (only required on computers running remote agent or server), define RT_FORCE_NODE_NAME as an environment variable and set its value to the FQDN of the computer.

For example, if the computer's FQDN is m1.company.com:

Windows

In Control Panel, System, Environment Variables, add, as a SYSTEM VARIABLE, the variable RT_FORCE_NODE_NAME with a value of m1.company.com.

UNIX

Add RT_FORCE_NODE_NAME=m1.company.com; export RT_FORCE_NODE_NAME to the USER PROFILE for every product user on the computer or the SYSTEM PROFILE.

Important! When RT_FORCE_NODE_NAME is defined on a remote agent computer, its value must be used as the agent's computer name for all product utilities that connect to the agent.

For example, if RT_FORCE_NODE_NAME is defined with the value of m1.company.com:

- Using the Workbench, log in to the agent using m1.company.com as the agent computer name.
- Using the hco command, check out a file to the remote agent using m1.company.com as the agent computer name.

  ```
  hco ... -rm m1.company.com ...
  ```

Important! When RT_FORCE_NODE_NAME is defined on a broker computer, its value must be used as the broker name for all product utilities that connect to the broker.

For example, if RT_FORCE_NODE_NAME is defined with the value of m1.company.com:

- Using the Workbench, log in to the broker using m1.company.com as the broker name.
- Using the hco command, check out a file using m1.company.com as the broker name.

  ```
  hco -b m1.company.com ...
  ```

The RTserver Port Number

The default RTserver communication port number is 5101. This port number can be set to any accessible port in your network. For example, if the default port of 5101 is not in the range of available ports across a firewall, you will need to specify another RTserver port number.

Note: Before changing the port number, you must stop the CA Harvest SCM broker, product clients, product agents, and the RTserver.

RTserver Setup

The RTserver startup command file must specify the Conn_Names option with the logical connection names value. The Conn_Names option specifies a list of logical connection names used by RTclient (the CA Harvest SCM client) to find this RTserver. Each logical connection name has the form protocol:node:address.

Note: The RTserver startup command file, rtserver.cm, is found in the RTHOME/standard directory.

Port number 5101 (default) represents the address portion of protocol:node:address. To specify a different port, define the Conn_Names option in the rtserver.cm file.

Example rtserver.cm file:

```
/* -------------------------------- */
/* RTserver-specific options */
/* -------------------------------- */
setopt prompt                   "SERVER> "
setopt time_format              hms
setopt client_connect_timeout   10.0
setopt client_max_buffer 10000000 /* ~10 MB */
setopt log_in_client            UNKNOWN
setopt log_in_server            UNKNOWN
setopt log_out_client           UNKNOWN
setopt log_out_server           UNKNOWN
setopt udp_broadcast_timeout    5.0
setopt Enable_Control_Msgs      connect, disconnect

setopt Conn_Names tcp:ca scm:5202
```

In this example, CA Harvest SCM specifies the computer node name.

You must restart the RTserver for the changes to take effect.

Specify the Network Port Number

The CA Harvest SCM client, broker, and server must specify the network port number using the -rtserver option in the following argument files: hbroker.arg, HServer.arg, and HClient.arg. The argument files are located in the CA_SCM_HOME directory.

Note: The argument file, HClient.arg, is not created by the installation and must be created manually in the CA_SCM_HOME directory. Other than specifying nondefault port numbers, and the option -cm[*i,a*] to specify the connect method for the connection between the client and the server, the HClient.arg file is not necessary.

Follow these steps:

1. Add the -rtserver option to the hbroker.arg, HServer.arg, and HClient.arg files using the following syntax:

   ```
   -rtserver=tcp:node:address
   ```

   For example:

   ```
   -rtserver=tcp:scm:5202
   ```

   The network port number is specified.

2. Restart the RTserver, in addition to the product's broker and clients.

   The changes take effect.

The Server Port Range

The CA Harvest SCM server uses operating system TCP/IP ports in addition to the RTserver TCP/IP communication port for the following functions: check-in, check-out, form file attachments, load repository and browsing the file system using a product remote agent.

The additional TCP/IP ports are assigned by the operating system dynamically (ephemeral ports) unless otherwise specified during the product server installation, or specified during post-installation using the instructions while specifying the CA Harvest SCM communication port number.

Configuring the TCP/IP port range to be used by the product is done completely on the product server installation. No product client configuration is necessary.

The product server port range is explicitly set in the RTserver startup command file, rtserver.cm for Windows-based product server installations and $RTHOME/standard/rtserver.cm for UNIX/Linux-based product server installations.

If the port range is not explicitly set in the rtserver.cm file, the product server uses operating system assigned ports (ephemeral ports) for the product functions mentioned in the previous section.

Define the Port Range

If you did not specify a TCP/IP port range during the CA Harvest SCM server installation, you can do this post-installation by performing the following procedure.

Important! The port range specified must be greater than or equal to the number of product server processes and remote agent processes running behind the firewall. If the port range is less than what is specified in the preceding paragraph, the check-in, check-out, load repository, form file attachment, and remote agent access may not function properly.

Follow these steps:

1. Open the following files for editing:
   - For Windows: %RTHOME%\standard\rtserver.cm
   - For UNIX/Linux: $RTHOME/standard/rtserver.cm
2. Define the option, setopt direct_connect_port_range, in the rtserver.cm file. For example, if the port range to specify is 9000 to 9100 inclusive:

   ```
   setopt direct_connect_port_range 9000,9100
   ```
3. Save and close the rtserver.cm file.

Note: You must restart RTserver, the product broker, and product clients for the changes to take effect.

Modify the Port Range

If you specified a TCP/IP port range during the CA Harvest SCM server installation and need to modify this port range post-installation, you can modify an existing port range by following the instructions in this procedure.

Important! The port range specified must be greater than or equal to the number of product server processes and remote agent processes running behind the firewall. If the port range is less than what is specified in the preceding paragraph, the check-in, check-out, load repository, form file attachment, and remote agent access may not function properly.

Follow these steps:

1. Open the following files for editing:
   - For Windows: %RTHOME%\standard\rtserver.cm
   - For UNIX/Linux: $RTHOME/standard/rtserver.cm

2. Locate the port range option, setopt direct_connect_port_range. Edit the range as necessary. For example, if the current range is 9000 to 9050 and you want to increase it to 9100, edit as follows:

   ```
   setopt direct_connect_port_range 9000,9050
   ```

   change to:

   setopt direct_connect_port_range 9000,9100

3. Save and close the rtserver.cm file.
4. (UNIX/Linux systems) Source the PEC environment script located in $RTHOME/bin/rtinit.sh(csh):

   . ./rtinit.sh (Bourne/Korn/Bash shells)

   source rtinit.csh (C shell)

5. Stop the RTserver process by executing the following command:

   ```
   rtserver -stop
   ```

Note: After the RTserver process has been stopped, the product broker automatically restarts the RTserver process. Stopping the RTserver will not interrupt product client operations.

Time-out Parameters

The time-out parameters affect the running of multiple concurrent CA Harvest SCM operations related to the product agent process. An example is a group of users that simultaneously check out files from the product to remote computers that run the product agent processes. The time-out parameters are:

- _*conn*_init_timeout
- ptm_client_accept_timeout
- ptm_server_accept_timeout

Due to system resource constraints and concurrency usage conflicts, sometimes a user may encounter a time-out error message. However, a user can change parameters to achieve better concurrency usage and avoid this time-out error.

The Connection Time-out Values

The default value for the RTserver connection time-out is five seconds. However, this value may be too short to accommodate all connection needs if a very large number of messages is sent to the RTserver simultaneously. Users can extend this time-out period by altering the PEC setting parameter _CONN_INIT_TIMEOUT in RTHOME/standard/rtclient.cm file on the computer that runs the product client process. See the following example for the increase of this time-out setting:

```
  /* increase to 30 seconds */
 setopt _conn_init_timeout 30.0
```

Increasing this time-out value may introduce side effects. For example, an invalid hostname specified to log in to a product agent computer can introduce the login time-out due to the message delivery failure. Users will wait much longer to receive the message regarding the connection failure. Therefore, users need to specify the time period based on the concurrency requirement.

If you receive an error on the standard output of the product command line client process similar to the following, you need to increase this time-out value:

```
WARNING: Attempted to read from connection client:local:<broker>:RTSERVER.
First read size is 8.  The number of bytes read returned by TipcLinkRecvTimeout is
-1.
WARNING: Last RTworks Error Code: 9
timeout reached
Last C Error Code: 2
No such file or directory
```

When the product performs the bulk data transfer (for example, check out a file), it uses peer-to-peer connection between client, server, and agent. Sometimes users may encounter a time-out error for establishing this peer-to-peer connection during a check-out in a high concurrency usage scenario. In the PEC install directory, RTHOME/standard/rtclient.cm file, the time-out parameters, ptm_client_accept_timeout and ptm_server_accept_timeout, can be changed to avoid time-out errors. The default value is 60 seconds for both. See the following examples to change the default values for these options.

Examples:

If you receive an agent login time-out error, “The login operation timed out,” specify the time-out value for ptm_client_accept_timeout in the RTHOME/standard/rtclient.cm file on the product client host computer:

```
/* increase to 120 seconds */
setopt ptm_client_accept_timeout 120.0
```

If you receive a network time-out error, “Network timeout,” specify the time-out value for ptm_server_accept_timeout in the RTHOME/standard/rtclient.cm file on the product server host computer:

```
/* increase to 120 seconds */
ptm_server_accept_timeout 120.0
```

Restart the product server after setting ptm_server_accept_timeout.

Chapter 16: Customizing the Product

Important! Customizing CA Harvest SCM is one step in the overall implementation process. To understand all of the steps you must complete for a successful implementation, see How to Implement the Product (see page 16).

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

Oracle International Language, Character Set, and Locale Settings

Important! This information applies only to non-English product installations.

The following sample settings apply to Oracle repositories only and are required for international Web Interface and CA Harvest SCM installations.

Note: For more information about how SQL Server supports non English character sets and locale settings, see your SQL Server documentation.

Check the following:

- Does Oracle support your language?
- What character set should you use for Oracle?

Several system tables and views use language settings in Oracle. Check these settings using the following SQL statements:

NLS_DATABASE_PARAMETERS

Default set up for the database at creation:

```
'select * from nls_database_parameters;'
```

NLS_INSTANCE_PARAMETERS

Parameters set in initSID.ora:

```
'select * from nls_instance_parameters;'
```

NLS_SESSION_PARAMETERS

Parameters set by NLS_LANG or alter session (does not include client character set):

```
'select * from nls_session_parameters;'
```

V$NLS_PARAMETERS

Values of NLS parameters for current values:

```
'select value from v$nls_parameters;'
```

V$NLS_VALID_VALUES

Values which can be used for CHARACTERSET, LANGUAGE, TERRITORY, and SORT (depending on what was loaded at installation time).

You may also want to adjust other Oracle NLS parameters such as NLS_CALENDER, NLS_DATE_FORMAT, and NLS_TERRITORY.

To help ensure that your extended ASCII characters such as the Euro sign (€,) and umlauts (for example, ä, ö, or, ü) are stored correctly in Oracle, create your database with a character set which includes the characters you want to display. For example, the Euro sign € requires set WE8ISO8859P15.

These settings apply to the Oracle server. All Oracle clients, including any product client computer, must have the corresponding NLS_LANG environment variable set correctly. Use the following syntax:

```
NLS_LANG = language_territory.character_set
```

language

Specifies the language and conventions used for displaying messages, day names, and month names.

territory

Specifies the territory and conventions used for calculating week and day numbers.

character_set

Controls the character set used for displaying messages.

Note: For more information about the sub-key locations for multiple Oracle homes, see the *Oracle Database Platform Guide for Windows*. For details about the NLS_LANG parameter and Globalization Support initialization parameters, see the *Oracle Database Globalization Support Guide*.

For example, to set this parameter for German on a UNIX or Linux computer, use the following:

```
NLS_LANG=GERMAN_GERMANY_WE8ISO8859P15
```

Then export the setting:

```
Export NLS_LANG
```

The following command is optional and displays the value for each login:

```
Echo NLS_LANG=$NLS_LANG
```

On Windows, set the NLS_LANG value in the user environment variables.

How to Modify Scripts or Profiles

UNIX or Linux users can perform the following tasks:

- Add the NLS_LANG and related variables to the wrapper scripts in the CA_SCM_HOME\bin directory.
- Add the NLS_LANG and related variables to their profiles.

Important! Users must perform one but not both of these tasks.

Modify Scripts

The CA_SCM_HOME/bin directory contains scripts that set up the environment for running the various CA Harvest SCM tools. These scripts are what are actually executed (not the product tools directly), because the scripts call the tools.

Follow these steps:

1. Edit the $CA_SCM_HOME/install/harvestapp.sh file.
2. Add the following lines to the file:

```
NLS_LANG=language.territory.character-set
export NLS_LANG
```

3. Save the file and exit.
4. Run the product installation program $CA_SCM_HOME/install/install.sh and select Option 3 for maintenance.
5. Answer all prompts and exit the program.

Modify Profiles

You can also add the following lines to your .profile file:

```
NLS_LANG=language.territory.character-set
export NLS_LANG
```

Note: For more information, see your operating system documentation.

Shell and Other Operating System Settings

The following information applies to all CA Harvest SCM components, including the server, client, and agent.

Review the following guidelines before changing any operating system settings.

- Code page is the name for the operating system encoding schemes, roughly the equivalent of the character set in Oracle.
- You must distinguish between a font and a character set or code page. A font is used by the operating system to convert a numeric value into a graphical character representation on screen.
- Identify the character set or code page used by your clients. Contact your operating system manufacturer (Microsoft, Hewlett-Packard, Sun, and so on) on how to obtain this information.

Note: For UNIX, the character set is also defined by the terminal emulation used, for example, X11.

The code page and fonts used by your operating system must support the national language settings defined within Oracle. Otherwise, the Web Interface and other client applications may be unable to insert the correct characters into the product repository, even if Oracle can store them correctly. If the settings do not match, a character conversion can take place. For example, the entered Euro sign could be converted into a question mark.

Note: You should change the configuration parameters of the workbench.ini file when any discrepancy with any of the languages for MS932 is observed, and provide alternate encoding styles.

An easy way to test the compliance of client and Oracle server settings is to log in to SQL*Plus in your operating system shell using your Oracle cascm user. At the SQL*Plus prompt, create a test table in the Oracle CASCM schema as follows:

```
DROP TABLE HARCHARTEST;

CREATE TABLE HARCHARTEST
    ("ITEMOBJID" NUMBER NOT NULL
    , "ITEMNAME" VARCHAR2(2000))
    TABLESPACE "HARVESTMETA" LOGGING;
```

Then insert some extended ASCII values into the table to test both your client shell and Oracle settings:

```
INSERT INTO HARCHARTEST( ITEMOBJID, ITEMNAME ) VALUES( 1, 'CharàääöTEST.TXT' );
```

Then display the inserted values:

```
select * from harchartest;
```

If the extended ASCII chars appear correctly in the result set, both your Oracle server and client shell settings should be correct.

Web Interface Settings

If your third-party language settings do not match those in the Web Interface configuration file (harweb.cfg), the Web Interface may be unable to display certain characters. This file can be found

- On Windows-..\harweb\WEB-INF\harweb.cfg
- On UNIX or Linux-../harweb/WEB-INF/harweb.cfg

To support all Latin characters except for the Euro sign, you need the following setting:

```
CharacterSet=iso-8859-15
```

Important! Do not use the Euro sign with the Web Interface. In addition, on Japanese it is important to keep Oracle, NLS_LANG, and the Web Interface character set in the same dialect (Kanji, Katakana, or Hiragana).

The Web Interface (CA Harvest SCM) operating system user must have the Oracle NLS_PARAMETERS settings in the user profile (.profile, or .bash_profile). If the Web Interface user is not the one running the servlet or web server, you must also verify that this user's variables include the $NLS_LANG and $LANG parameters as described previously in Oracle Settings.

It may be necessary to modify the default ISO-8859-1 encoding for some languages, such as Japanese or German, in the following files:

- On Windows-..\harweb\WEB-INF\web.xml
- On UNIX or Linux-../harweb/WEB-INF/web.xml

Custom form types for Harweb use must not include any non-English characters for the following elements; otherwise, forms that are created of that particular form type will not open in Harweb.

- Form type name
- Form type database table name
- Labels/names of created elements on the form type
- Database column names of created elements on the form type

Servlet or Web Server Settings

Contact your vendor to help ensure that your servlet or web server fully supports your language settings and code page.

Workbench Compatibility with Database Character Encoding

The CA Harvest SCM Workbench is based on Java. For Workbench to communicate with a CA Harvest SCM server, the Java System Property file.encoding must be compatible with the encoding of the CA Harvest SCM database.

To determine the encoding the Workbench uses, follow these steps:

1. Launch Help, About CA Harvest SCM Workbench.

   The About dialog appears.

2. Click the System Info button.

   The Configuration Details dialog appears.

3. Scroll down about 30 lines and the file.encoding setting is listed, for example, file.encoding=ISO-8859-P1.

Workbench on Windows

On Western language Windows systems, file.encoding is typically set to CP1252, a Windows encoding compatible with ISO 8859-1 and 8859-15. Typically, this encoding should be compatible with a Western language CA Harvest SCM database.

Workbench on Linux and zLinux

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

On Linux systems, character encoding varies depending upon the system's locale settings. Linux distributions typically use UTF-8 as their default encoding. If your installation uses UTF-8 as the default character encoding for its locale, you can use one of the following methods to make the Workbench compatible with your CA Harvest SCM server:

- Change your entire Linux system to use a legacy (non-UTF-8) character encoding.

  This method is the easiest to implement. A single change to the operating system allows all CA Harvest SCM clients to work properly by default, but it may have side-effects on other applications.

- Make changes to the startup configuration for the Workbench so that it is started with character encoding settings that match the CA Harvest SCM database.

  If you select this method, the CA Harvest SCM agent expects files on your file system to be named with a character encoding that matches your database. Files checked-out from CA Harvest SCM will use a legacy (non-UTF-8) character encoding for their names on the file system, even though other files on your file system are likely to use UTF-8 encoding for their names. New files being checked in to CA Harvest SCM must also have names encoded in a legacy (non-UTF-8) character encoding.

The method that you select depends upon your environment and needs.

Workbench on Linux and zLinux: Change UTF-8 System to Use a Legacy Encoding

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

You can change your system to use legacy (non-UTF-8) character encoding. This is the easiest and most reliable way to run Workbench on a Linux system. However, this will alter the default encoding for all applications on your Linux system, so it may not be desirable in your environment.

The procedure to make this change varies by system. This example shows how to change a SUSE Linux ES 10 system to use a legacy (non-UTF-8) character encoding.

1. Launch the SUSE YaST Control Center and click the Language icon (in the System category).

   The Primary Language Settings dialog appears.

2. Click Details and clear the Use UTF-8 Encoding option. Click OK.

3. Click Accept to apply the changes in YaST.
4. Log out of your desktop, and then log back in.
5. Check the LANG variable value.

   UTF-8 encoding is no longer specified. All applications will default to a legacy character encoding. On German systems this is typically a locale of de_DE@euro which translates to an encoding of IS0-8859-15.

Workbench on Linux and zLinux: Run Workbench on a UTF-8 System

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

If you want your system to remain configured with UTF-8 as its default encoding, the locale for the environment from which Workbench is launched must be configured with a non-UTF-8 character encoding. To do this, modify the LANG environment variable in the environment that starts Workbench as shown in the following example.

Note: Verify that before starting the Workbench, the Linux Terminal Character Encoding is set to the correct locale. In a GNOME terminal, you can change the Terminal Character Encoding from Terminal, Set Character Encoding menu option.

Example: Modify Workbench to Work on a UTF-8 System

This example shows how to modify the Workbench on a German system. The Workbench can be started using the following command:

```
export LANG=de_DE@euro
./workbench
```

When you start the Workbench this way, the Java file.encoding system property is automatically set by the Java VM based upon the locale, which in this case is ISO-8859-15. You do not need to specify a file.encoding setting in the workbench.ini file; it automatically inherits its setting from the locale.

The CA Harvest SCM Agent will expect files on your file system to be named with a character encoding that matches your database. Files checked-out from CA Harvest SCM will use a legacy (non-UTF-8) character encoding for their names on the file system, even though other files on your file system are likely to use UTF-8 encoding for their names. New files being checked in to CA Harvest SCM must also have names encoded in a legacy (non-UTF-8) character encoding. Because of this, you may find it necessary to change the filename encoding of existing files on your file system. On Linux systems, a common way of changing this encoding is to use the Linux convmv script. This script is shipped with SUSE Linux ES 10.

Web Server on UNIX

On UNIX systems, the Java System Property file.encoding varies depending upon the system's locale settings. Most Linux distributions use UTF-8 as their default encoding. If file.encoding is set to UTF-8, use one of the following methods to make Java file.encoding compatible with the CA Harvest SCM database encoding:

- Change your UNIX system to use a legacy (non-UTF-8) character encoding.

  This method is the easiest one. A single change to the operating system allows all CA Harvest SCM clients to work properly by default, but it may have side-effects on other applications. The procedure to make this change will vary by system.

- Change the startup configuration for your Web Server so that it is started with character encoding settings that match the CA Harvest SCM database.

  An easy way to alter file.encoding on UNIX systems is to change the LANG variable in the environment from which Tomcat is started.

Example: Alter Locale When Starting Tomcat

This example shows how to start Tomcat on a UNIX system with a locale that matches the CA Harvest SCM database character encoding. In this example, Tomcat is started on a German UNIX system as follows:

```
export LANG=de_DE@euro
./startup.sh
```

The Java VM initializes file.encoding based upon the locale specified by LANG. In this example, file.encoding is initialized automatically to the value ISO-8859-15. If LANG is set to de_DE, file.encoding is initialized to ISO-8859-1.

Command-Line Tools

The shell in which command-line tools are executed must use a character encoding that is compatible with your database.

If you are using a UNIX system that is configured to use UTF-8 as its default encoding, command-line tools must be executed in a modified command shell that is using a character encoding compatible with server. The method to change the character encoding of your command shell will differ by system.

As an example, for Linux environments running Gnome, the gnome-terminal application offers selection of the character encoding on its main menu. For Western languages, this would typically be ISO-8859-1 or ISO-8859-15.

An alternative approach is to change your entire Linux system to not use UTF-8 encoding (see page 361).

Software Development Kit (SDK) Components

CA Harvest SCM includes a Software Development Kit (HSDK), a Java Software Development Kit (JHSDK), and a Component Object Model SDK (COM SDK) that you can use to build applications. To create and build JHSDK applications, the JHSDK must be installed locally. To run JHSDK applications, the JHSDK, HSDK, and the CA Harvest SCM client must be installed locally.

- On Windows, when you install the server, client, and agent, these SDK components and automatically installed.
- On UNIX and Linux, when you install the client (command-line utilities), these SDK components are automatically installed.

The JHSDK and HSDK files included for UNIX and Linux are the same files included for Windows. However, as the following information illustrates, the library names are slightly different on UNIX and Linux.

Platform	Library Names
Solaris and Linux	$CA_SCM_HOME/HSDK/lib/libHSDK.so $CA_SCM_HOME/lib/libHSDK.so $CA_SCM_HOME/lib/libJHSDK.so $CA_SCM_HOME/JHSDK/lib/jhsdk.jar
HP-UX	$CA_SCM_HOME/HSDK/lib/libHSDK.sl $CA_SCM_HOME/lib/libHSDK.sl $CA_SCM_HOME/lib/libJHSDK.sl $CA_SCM_HOME/JHSDK/lib/jhsdk.jar
AIX	$CA_SCM_HOME/HSDK/lib/libHSDK.a $CA_SCM_HOME/lib/libHSDK.a $CA_SCM_HOME/lib/libJHSDK.a $CA_SCM_HOME/JHSDK/lib/jhsdk.jar

Note: As previously illustrated, the libHSDK.ext files are installed in two locations ($CA_SCM_HOME/HSDK/lib/, to place the HSDK in its own self-contained directory structure, and $CA_SCM_HOME/lib/, to place all libraries required by the JHSDK in one location). This provides maximum flexibility when using the HSDK.

After installation, you must set up and test the SDK components. You can then use them to build applications.

Note: For information about how to use the HSDK, JHSDK, and COM SDK, see the *SDK Reference Guide*.

How to Set Up and Test the SDK Components

After you install the server, client, client (command-line utilities), and agent, and if you want to use the SDK components, you must set up and test the SDK components. You can then use them to build applications. To successfully set up and test the SDK components, complete the following steps:

1. Set up a development environment.
2. Test the HSDK installation.
3. Test the HSDK and JHSDK installation.
4. Set up the product to use the JHSDK programs.

More information:

How to Set Up a Development Environment (Windows) (see page 365)
Set Up a Development Environment for the Command-Line Utilities (UNIX, Linux, and zLinux) (see page 366)
Test the HSDK Installation (Windows) (see page 367)
Test the HSDK and JHSDK Installation (UNIX, Linux, and zLinux) (see page 367)
Set Up the Product to Use the JHSDK Programs (Windows) (see page 369)
Set Up the Product to Use the JHSDK Programs (UNIX, Linux, and zLinux) (see page 370)

How to Set Up a Development Environment (Windows)

The first step you must complete for the Software Development Kit (SDK) components is to set up a development environment. You must complete this step before you can test that the SDK was installed successfully, and before you can create and run SDK programs.

Follow these steps:

1. Place the HSDK\include directory in the include path for your development tool.
2. Place the HSDK\lib directory in the library path for your development tool.
3. Include hsdk.h in any application file that uses HSDK classes.
4. Include hsdk.lib in the link command for the application.
5. Run the HSDK sample test program.

The HSDK supplies a sample program and the Microsoft Developer Studio project files to build and run the test program. The sample project was established with Developer Studio and Service Pack 3. The project file sets the proper include and lib paths for the build.

The sample program requires that a specific CA Harvest SCM project exists in the product repository. The SDKSampleProject.har file is an export file of the required project.

Set Up a Development Environment for the Command-Line Utilities (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The first step you must complete for the Software Development Kit (SDK) components is to set up a development environment. You must complete this step before you can test that the SDK was installed successfully, and before you can create and run SDK programs.

Follow these steps:

1. Add the HSDK/include directory to the include path for your development tool.
2. Add the HSDK/lib, JHSDK/lib, and $RTHOME/lib/$RTARCH to the library path for your development tool.
3. Include hsdk.h in any application file that uses HSDK classes.
4. Include the following libraries in the link command for the application: -lHSDK -lhcomm -lsignfile -lharagent -lhauth -lhapi -lhutils -lcxxipc -lrtutil -lrtipc -lrtptm –lrtzlib. The following table illustrates how the libraries are linked:

Directory	Libraries
$CA_SCM_HOME/lib	HSDK JHSDK hcomm signfile haragent hauth hapi hutils
/opt/CA/SharedComponents/CAPKI(customizable)	capki
$RTHOME/lib/$RTARCH	-lcxxipc -lrtutil -lrtipc -lrtptm -lrtcrypt -lrtzlib

Test the HSDK Installation (Windows)

The next step you must complete for the Software Development Kit (SDK) components is to test the HSDK installation to verify that it was installed successfully and that the SDK is ready to be set up.

Follow these steps:

1. Run himpenv.exe to import the SDKSampleProject.har file.
2. Create the user *sdkuser* with the password *sdkpass*.
3. Add the user to the SDKSampleProject user group and to the approval list on the Approve process in state-1.
4. Create a read/write repository named \sdkrep and assign it to the baseline view of SDKSampleProject.
5. Create the directory C:\temp\hsdkdir.
6. In Developer Studio, open HSDK\Samples\hsdksample.dsw.
7. Build the project.
8. Set the program arguments to -b brokername.
9. Run the program.
10. Check the file HSDKSample.log for the results.

Test the HSDK and JHSDK Installation (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

The next step you must complete for the Software Development Kit (SDK) components is to test the HSDK and JHSDK installations to verify that they were installed successfully and that the SDK is ready to be set up. These steps involve running the HSDK or JHSDK sample test program. The HSDK includes a sample test program (hsdksample.cpp), the makefile (hsdksample.mak), and shell script (hsdksample.sh) to build it. The hsdksample.sh file sets the proper include and library paths for the build.

The sample program requires the existence of the CA Harvest SCM project named SDKSampleProject in the product repository. The SDKSampleProject.har file is a product export file of the required project.

Follow these steps:

1. If you have not already done so, set up the library path and system variables.

   Note: For information about setting up the library path and system variables, see Set Up a Development Environment for the Command-line utilities (see page 366).

2. Load the script for initializing the Enterprise Communicator (PEC) for use with the HSDK and JHSDK by running one of the following commands from the /bin subdirectory in the PEC installation directory:

   - If you are using the Bourne, Korn, or Bash shell, run the following command:

     ```
     . ./rtinit.sh
     ```

   - If you are using the C shell, run the following command:

     ```
     source rtinit.csh
     ```

3. Verify that RTHOME is set by running the following echo command:

   ```
   echo $RTHOME
   ```

4. Run the sample test program by successfully completing these steps:

   a. (AIX users only). In the hsdksample.sh file, in the AIX section, replace the stlport-4 6.2 references with the complete path name in which stlport 4.6.2 is installed.

   b. Run himpenv to import the SDKSampleProject.har file.

   c. Create the user *sdkuser* with the password *sdkpass*.

   d. Add the sdkuser to the SDKSampleProject user group and to the approval list on the Approve process in state-1.

   e. Create a read/write repository named \sdkrep and assign it to the baseline view of SDKSampleProject.

   f. The directory HSDK/Release contains the compiled version of the sample program. Remove the compiled version from the directory so you can rebuild the sample program.

   g. Change directory (cd) to the Samples directory.

   h. (Red Hat Linux users only). Enter the following command:

      ```
      export CCOPTS=-DRHEL_AS3
      ```

   i. To rebuild the sample program, run the hsdksample.sh script with the target RELEASE:

      ```
      ./hsdksample.sh RELEASE
      ```

   j. Check the HSDK/Release directory for the newly rebuilt sample program.

k. Execute the sample program in HSDK/Release, by entering the *./hsdksample -b <brokername>* command.

l. Check the HSDKSample.log file for the results.

The following is a typical example for setting up the HSDK and JHSDK environment on a Solaris system:

```
CA_SCM_HOME=/opt/CA/scm
. /opt/CA/pec/bin/rtinit.sh #Sets $RTHOME, $RTARCH, LD_LIBRARY_PATH
LD_LIBRARY_PATH=/usr/local/ETPKI:$LD_LIBRARY_PATH
# Uncomment below and edit for custom ETPKI path that is the
# alternative to /usr/local/ETPKI, such as /opt/CA/ETPKI
LD_LIBRARY_PATH=${CA_SCM_HOME}/lib:$LD_LIBRARY_PATH
PATH=${CA_SCM_HOME}/bin:${PATH}
export CA_SCM_HOME LD_LIBRARY_PATH PATH
```

Set Up the Product to Use the JHSDK Programs (Windows)

Before you can use JHSDK sample programs, you must complete the following steps for the CA Harvest SCM repository and the client computer on which you will build and run JHSDK programs.

Follow these steps:

1. Execute himpenv.exe to import the JHSDKSampleProject.har file.
2. Create the Windows operating system user named *sdkuser* with the password *sdkpass*.

   Note: These logon credentials are used in the sample JHSDK commands that are used by the sample code. For information, see the *SDK Reference Guide*.
3. Add sdkuser to the JHSDKSampleProject user group and to the approval list on the Approve process in state-1.
4. Create a read/write repository named *sdkrep* and assign it to the baseline view of JHSDKSampleProject.
5. Create the following directory, if it does not exist already:

   ```
   C:\temp\jhsdk_sample
   ```
6. In the directory created in the previous step, create a text file named jhsdk_sample_file.txt, if it does not exist already.
7. Enter and save a small amount of text in the jhsdk_sample_file.txt file.
8. Change to the directory %CA_SCM_HOME%\JHSDK\samples\com\ca\harvest\jhsdk\sample and enter the following command to create .class files:

   ```
   javac -classpath %CA_SCM_HOME%\jhsdk.jar *.java
   ```

9. Check this directory and verify that every .java file corresponds to a matching .class file. For example, the Sample1.java file corresponds to the Sample1.class file, Sample2.java corresponds to Sample2.class, Sample3.java corresponds to Sample3.class, and so forth.
10. Use the following command to set the classpath variable:

    ```
    SET CLASSPATH=%CLASSPATH%;.
    ```

    Note: Verify that you include the ending semicolon (;) and period (.) in this command. This adds the current directory to the classpath. In addition, if you modify the default installation location of the Java files, you must modify the classpath variable accordingly. Finally, if you do not set the classpath, you may receive error messages indicating that the class cannot be found.
11. Change to the %CA_SCM_HOME%\JHSDK\samples directory and use the following command to run the sample programs:

    ```
    java com.ca.harvest.jhsdk.sample.SampleRunner -b broker-name
        -u sdkuser -p  sdkpass -s n
    ```

    Note: In this command, *broker-name* defines the name of your broker computer, and *n* specifies the number of the sample program to run. For example, enter 1 to run sample program 1.
12. Review the source code comments supplied with the JHSDK, especially the Requirements section.

Set Up the Product to Use the JHSDK Programs (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

Before you can use the JHSDK sample programs, you must complete the following steps for the CA Harvest SCM repository and the client computer on which you will build and run JHSDK programs. The setup steps include an export file for the JHSDK samples. The export file works like the HSDK lifecycle export file. The export file includes the Test UDP user-defined process required to run the JHSDK sample programs.

Follow these steps:

1. If you have not already done so, run himpenv to import the JHSDKSampleProject.har file.
2. If you have not already done so, create the user *sdkuser* with the password *sdkpass*.
3. If you have not already done so, add sdkuser to the JHSDKSampleProject user group and to the approval list on the Approve process in state-1.
4. If you have not already done so, create a read/write repository named *sdkrep* and assign it to the baseline view of JHSDKSampleProject.

5. Edit the .java sample files and replace the occurrences of c:\\temp\\jhsdk_sample with a directory on your local system.
6. Create the following directory, if it does not exist already:

   ```
   C:\temp\jhsdk_sample
   ```

7. In the directory created in the previous step, create a text file named jhsdk_sample_file.txt, if it does not exist already.
8. Enter and save a small amount of text in the jhsdk_sample_file.txt file.
9. Change to the directory $CA_SCM_HOME/JHSDK/samples/com/ca/harvest/jhsdk/sample and enter the following command to create .class files:

   ```
   javac -classpath $CA_SCM_HOME/JHSDK/lib/jhsdk.jar *.java
   ```

10. Check this directory and verify that every .java file corresponds to a matching .class file. For example, the Sample1.java file corresponds to the Sample1.class file, Sample2.java corresponds to Sample2.class, Sample3.java corresponds to Sample3.class, and so forth.
11. Set the CLASSPATH variable.

    - If you are using the Bash or Bourne shell, enter the following commands:

      ```
      CLASSPATH=$CLASSPATH:.
      export CLASSPATH
      ```

    - If you are using the C shell, enter the following command:

      ```
      setenv CLASSPATH $CLASSPATH:.
      ```

    Note: Verify that you include the ending colon (:) and period (.) in this command. This adds the current directory to the classpath. In addition, if you modify the default installation location of the Java files, you must modify the classpath variable accordingly. Finally, if you do not set the classpath, you may receive error messages indicating that the class cannot be found.

12. Change to the $CA_SCM_HOME/JHSDK/samples directory and use the following command to run the sample programs:

    ```
    java com.ca.harvest.jhsdk.sample.SampleRunner -b broker-name
        -u sdkuser -p  sdkpass -s n
    ```

    Note: In this command, *broker-name* defines the name of your broker computer, and *n* specifies the number of the sample program to run. For example, specify 1 to run sample program 1.

    Note: For information, see the source code comments in the JHSDK Java sample files.

Chapter 17: Uninstalling on Windows

This section contains the following topics:

Uninstall the Server

You may decide to uninstall the CA Harvest SCM server for various reasons. For example, you may need to move the server to a different location on a different computer, or when you need to manually upgrade the server. When you uninstall the server, your existing product database is *not* removed.

Important! To use the latest release of the product component, you must first uninstall the existing component release. If you use the Upgrade Wizard, this step is automatically done for you. However, when you do not use the Upgrade Wizard, you must manually perform this step before installing the new release of the product component.

Follow these steps:

1. Verify that all product programs, services, and integrated development environments (IDEs) are closed.
2. In the Control Panel (Add or Remove Programs), run the server uninstallation program.

   The server is uninstalled.

Uninstall the Client

You may decide to uninstall the CA Harvest SCM client for various reasons.

Important! To use the latest release of the product component, you must first uninstall the existing component release. If you use the Upgrade Wizard, this step is automatically done for you. However, when you do not use the Upgrade Wizard, you must manually perform this step before installing the new release of the product component.

Follow these steps:

1. Verify that all product programs, services, and integrated development environments (IDEs) are closed.
2. In the Control Panel (Add or Remove Programs), run the client uninstallation program.

   The client is uninstalled.

Uninstall the Agent

You may decide to uninstall the CA Harvest SCM agent for various reasons.

Important! To use the latest release of the product component, you must first uninstall the existing component release. If you use the Upgrade Wizard, this step is automatically done for you. However, when you do not use the Upgrade Wizard, you must manually perform this step before installing the new release of the product component.

Follow these steps:

1. Verify that all product programs, services, and integrated development environments (IDEs) are closed.
2. In the Control Panel (Add or Remove Programs), run the agent uninstallation program.

   The agent is uninstalled.

Uninstall CA Harvest SCM Reports

You may decide to uninstall CA Harvest SCM Reports for various reasons.

To uninstall CA Harvest SCM Reports, in the Control Panel (Add or Remove Programs), run the CA Harvest SCM Reports uninstallation program.

CA Harvest SCM Reports is uninstalled.

Note: For information about uninstalling CA Business Intelligence from Windows, see the *CA Business Intelligence Implementation Guide*.

Chapter 18: Uninstalling on UNIX, Linux, and zLinux

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

How to Uninstall the Server and Related Components

Important! CA Harvest SCM and several other CA products use these shared components. When you uninstall these components, you may adversely affect other locally-installed CA products that use them. Therefore, before you remove any shared component, contact your system administrator. In addition, verify that all of the component's processes are terminated.

You may decide to uninstall the CA Harvest SCM for various reasons. To uninstall the server and related components, complete these steps:

1. Uninstall the CA Harvest SCM server.
2. Uninstall CA Licensing (Lic98).
3. Uninstall the Public Key Infrastructure (eTPKI).
4. Uninstall CAI/PT ODBC.
5. Uninstall Enterprise Communicator (PEC).

More information:

Uninstall the Server

When you uninstall the CA Harvest SCM server locally, the local product agent and product command line utilitiescommand-line utilities Therefore, when you uninstall the product server locally, you do not need to perform any additional steps to uninstall the local product agent or product command-line utilities.

Follow these steps:

1. Back up the HBroker.arg, HServer.arg, or HAgent.arg files in the CA_SCM_HOME directory, if they have been customized.

   The default location for CA_SCM_HOME is /opt/CA/scm. To remove this directory, run the following command. If you used a custom installation directory instead of the default installation directory, replace the default with your custom location.

   ```
   rm -rf /opt/CA/scm
   ```

2. Delete the $CA_SCM_HOME directory. This directory stores the bkrd, hserver, agentd, and command-line utilities. Typically, the subdirectories in this directory are bin, lib, install, and log.

   The server is uninstalled.

Uninstall CA Licensing (Lic98)

By default, when you install CA Licensing, the $CASHCOM/ca_lic and $CASHCOMP/lib directories are created to contain the Lic98 files. You can optionally change this location.

Note: For information about changing the default CA Licensing directory, see Install Lic98 Licensing (see page 82).

The directory /usr/local/CAlib is shared. Therefore, before removing it, verify that no other locally installed CA products requires the component. In addition, before uninstalling CA Licensing (Lic98), back up the ca.olf file (your license key file).

Follow these steps:

1. Log in as a root user.
2. Run one of the following commands:

   ```
   rm -rf /opt/CA/ca_lic
   ```

   ```
   rm -rf /opt/CA/CAlib
   ```

   CA Licensing is uninstalled.

Uninstall the Public Key Infrastructure (CAPKI)

By default, the Public Key Infrastructure (CAPKI) is installed to /opt/CA/SharedComponents/ETPKI/lib. When necessary, you can uninstall CAPKI.

Important! CAPKI is a shared component. Therefore, before you remove it, verify that no other locally installed CA products require the component.

To uninstall the Public Key Infrastructure (CAPKI)

1. Log is as a root user or the owner of the CAPKI installation directory.
2. Run the following command:

   ```
   setup.exe remove caller="CA SCM" env=user
   ```

   ```
   The Public Key Infrastructure (CAPKI) is uninstalled.
   ```

Uninstall CAI/PT ODBC

By default, CAI/PT ODBC is installed to /opt/CA/caiptodbc (your $ODBC_HOME), but you can optionally specify a custom installation location. CAODBC provides an uninstallation utility that includes instructions for both automated and manual uninstallations.

CAODBC is a shared CA component. Therefore, before you remove it, verify that no other locally installed CA products requires the component. In addition, back up your $ODBC_HOME/odbc.ini file. This file is configured for all of the datasources used by CAODBC.

Follow these steps:

1. Log in as a root user or any user who owns the installation directory.
2. Verify that the $ODBC_HOME/uninstallation directory exists.
3. Run one of the following commands to load the CAODBC script.
   - For the Bourne shell, run the following command:

     ```
     . /opt/CA/caiptodbc/odbcenv.sh
     ```

   - For the C shell, run the following commands:

     ```
     source /opt/ca/caiptodbc/odbcenv.csh
     source /opt/CA/caiptodbc/odbcenv.csh
     ```

4. Run the following uninstallation script.

   ```
   $ODBC_HOME/uninstall/uninstall.sh
   ```

   CAI/PT ODBC is uninstalled.

Note: To manually uninstall CAI/PT ODBC, run the *rm -rf /opt/CA/caiptodbc* command and verify that you specify the installation directory correctly.

Uninstall Enterprise Communicator (PEC)

By default, the Enterprise Communicator (PEC) is installed to /opt/CA/pec (your $RT_HOME), but you can optionally specify a custom installation location. PEC provides an uninstallation utility that includes instructions for both automated and manual uninstallations.

PEC is a shared CA component. Therefore, before you remove it, verify that no other locally installed CA products require the component. In addition, back up your $RTHOME/standard directory, especially if the .cm files were customized.

To uninstall the Enterprise Communicator (PEC)

1. Log in as a root user or any user who owns the installation directory.
2. Verify that the $PEC/uninstallation directory exists.
3. Run one of the following commands to load the PEC environment script.
   - For the Bourne shell, run the following command:

     ```
     . /opt/CA/pec/bin/rtinit.sh [Bourne shell]
     ```
   - For the C shell, run the following command:

     ```
     source /opt/CA/pec/rtinit.csh [csh shell]
     ```
4. Run the following uninstallation script:

   ```
   $RTHOME/uninstall/uninstall.sh
   ```

Note: To uninstall PEC manually, run the rm -rf /opt/CA/pec command and make sure to specify the installation directory correctly.

How to Uninstall the Command Line Utilities and Related Components

Important! CA Harvest SCM and several other CA products use these shared components. When you uninstall these components, you may adversely affect other locally-installed CA products that use them. Therefore, before you remove any shared component, contact your system administrator. In addition, verify that all of the component's processes are terminated.

The CA Harvest SCM command-line utilities include the command-line utilities component, the Public Key Infrastructure (eTPKI), and the Enterprise Communicator (PEC). If you have already uninstalled the product server locally, the local command-line utilities were also uninstalled during that process. Therefore, you do not need to perform any additional steps to uninstall the local command-line utilities.

Follow these steps:

1. Uninstall the command-line utilities.
2. Uninstall the Public Key Infrastructure (eTPKI).
3. Uninstall Enterprise Communicator (PEC).

More information:

Uninstall the Command Line Utilities (see page 381)
Uninstall the Public Key Infrastructure (CAPKI) (Command Line Utilities) (see page 381)
Uninstall the Enterprise Communicator (PEC) (see page 382)

Uninstall the Command Line Utilities

The CA Harvest SCM command-line utilities include the command-line utilities component, the Public Key Infrastructure (eTPKI), and the Enterprise Communicator (PEC). If you have already uninstalled the product server locally, the local command-line utilities were also uninstalled during that process. Therefore, you do not need to perform any additional steps to uninstall the local command-line utilities.

To uninstall the command-line utilities, run the *rm -rf /opt/CA/scm* command to delete the $CA_SCM_HOME home directory. This directory stores the command-line utilities. Typically, the subdirectories in this directory are bin, lib, install, and log.

Note: The default $CA_SCM_HOME location is /opt/CA/scm. If you used a custom installation directory instead of the default installation directory, replace the default with your custom location.

Uninstall the Public Key Infrastructure (CAPKI) (Command Line Utilities)

By default, the Public Key Infrastructure (CAPKI) is installed to /opt/CA/SharedComponents/ETPKI/lib. When necessary, you can uninstall CAPKI.

Important! CAPKI is a shared component. Therefore, before you remove it, verify that no other locally installed CA products require the component.

To uninstall the Public Key Infrastructure (CAPKI)

1. Log is as a root user or the owner of the CAPKI installation directory.
2. Run the following command:

```
setup.exe remove caller="CA SCM" env=user

The Public Key Infrastructure (CAPKI) is uninstalled.
```

Uninstall the Enterprise Communicator (PEC)

By default, the Enterprise Communicator (PEC) is installed to /opt/CA/pec (your $RT_HOME), but you can optionally specify a custom installation location. PEC provides an uninstallation utility that includes instructions for both automated and manual uninstallations.

PEC is a shared CA component. Therefore, before you remove it, verify that no other locally installed CA products require the component. In addition, back up your $RTHOME/standard directory, especially if the .cm files were customized.

To uninstall the Enterprise Communicator (PEC)

1. Log in as a root user or any user who owns the installation directory.
2. Verify that the $PEC/uninstallation directory exists.
3. Run one of the following commands to load the PEC environment script.
   - For the Bourne shell, run the following command:

     ```
     . /opt/CA/pec/bin/rtinit.sh [Bourne shell]
     ```
   - For the C shell, run the following command:

     ```
     source /opt/CA/pec/rtinit.csh [csh shell]
     ```
4. Run the following uninstallation script:

   ```
   $RTHOME/uninstall/uninstall.sh
   ```

Note: To uninstall PEC manually, run the rm -rf /opt/CA/pec command and make sure to specify the installation directory correctly.

How to Uninstall the Agent and Related Components

Important! CA Harvest SCM and several other CA products use these shared components. When you uninstall these components, you may adversely affect other locally-installed CA products that use them. Therefore, before you remove any shared component, contact your system administrator. In addition, verify that all of the component's processes are terminated.

If you have already uninstalled the CA Harvest SCM server locally, the local product agent was also uninstalled during that process. Therefore, you do not need to perform any additional steps to uninstall the local agent.

Follow these steps:

1. Uninstall the agent.
2. Uninstall the Public Key Infrastructure (eTPKI).
3. Uninstall Enterprise Communicator (PEC).

More information:

Uninstall the Agent (see page 383)
Uninstall the Public Key Infrastructure (CAPKI) (Agent Installation UNIX and Linux) (see page 383)
Uninstall Enterprise Communicator (PEC) (see page 384)

Uninstall the Agent

If you have already uninstalled the CA Harvest SCM server locally, the local product agent was also uninstalled during that process. Therefore, you do not need to perform any additional steps to uninstall the local agent.

Follow these steps:

1. Back up the HAgent.arg file if you customized it.
2. Run the *rm -rf /opt/CA/scm* command to delete the $CA_SCM_HOME home directory. This directory stores the agntd directory and the subdirectories bin, lib, install, and log.

Note: The default $CA_SCM_HOME location is /opt/CA/scm. If you used a custom installation directory instead of the default installation directory, replace the default with your custom location.

Uninstall the Public Key Infrastructure (CAPKI) (Agent Installation UNIX and Linux)

By default, the Public Key Infrastructure (CAPKI) is installed to /opt/CA/SharedComponents/ETPKI/lib. When necessary, you can uninstall CAPKI.

Important! CAPKI is a shared component. Therefore, before you remove it, verify that no other locally installed CA products require the component.

To uninstall the Public Key Infrastructure (CAPKI)

1. Log is as a root user or the owner of the CAPKI installation directory.
2. Run the following command:

   ```
   setup.exe remove caller="CA SCM" env=user
   ```

   The Public Key Infrastructure (CAPKI) is uninstalled.

Uninstall Enterprise Communicator (PEC)

By default, the Enterprise Communicator (PEC) is installed to /opt/CA/pec (your $RT_HOME), but you can optionally specify a custom installation location. PEC provides an uninstallation utility that includes instructions for both automated and manual uninstallations.

PEC is a shared CA component. Therefore, before you remove it, verify that no other locally installed CA products require the component. In addition, back up your $RTHOME/standard directory, especially if the .cm files were customized.

To uninstall the Enterprise Communicator (PEC)

1. Log in as a root user or any user who owns the installation directory.
2. Verify that the $PEC/uninstallation directory exists.
3. Run one of the following commands to load the PEC environment script.
   - For the Bourne shell, run the following command:

     ```
     . /opt/CA/pec/bin/rtinit.sh [Bourne shell]
     ```

   - For the C shell, run the following command:

     ```
     source /opt/CA/pec/rtinit.csh [csh shell]
     ```

4. Run the following uninstallation script:

   ```
   $RTHOME/uninstall/uninstall.sh
   ```

Note: To uninstall PEC manually, run the rm -rf /opt/CA/pec command and make sure to specify the installation directory correctly.

Chapter 19: Uninstalling on z/OS

This section contains the following topics:

Uninstall the Agent

Use the following procedure to uninstall the Z/OS agent.

Follow these steps:

1. Back up the HAgent.arg file if you customized it.
2. Run the *rm -rf /opt/CA/scm* command to delete the $CA_SCM_HOME home directory. This directory stores the agntd directory and the subdirectories bin, lib, install, and log.

Note: The default $CA_SCM_HOME location is /opt/CA/scm. If you used a custom installation directory instead of the default installation directory, replace the default with your custom location.

Chapter 20: Uninstalling the Web Interface

This section contains the following topics:

Uninstall the Web Interface (Windows)

You may decide to uninstall the Web Interface for various reasons. To uninstall the Web Interface on JBoss, delete the harweb.war file directory from *%JBOSS_HOME%**server**default*\deploy or %JBOSS_HOME%\server*all*\deploy. To uninstall the Web Interface on Tomcat, delete the folder under the C:\Tomcat\webapps directory.

Note: For information about how to remove applications from Oracle iPlanet Web Server or WebSphere, see your application server's documentation.

Uninstall the Web Interface (UNIX, Linux, and zLinux)

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

On UNIX and Linux, the Web Interface is also comprised of the CA Harvest SCM command line utilitiescommand-line utilitiesinstall the Web Interface for various reasons. To uninstall the Web Interface on JBoss, delete the harweb.war file or directory from $JBOSS_HOME/server/*default*/deploy or $JBOSS_HOME/server/*all*/deploy. To uninstall the Web Interface on Tomcat, delete the /tomcat/webapps directory.

Note: For information about how to remove applications from Oracle iPlanet Web Server or WebSphere, see your application server's documentation.

Chapter 21: Configuring SCM for CA Vision Integration

Configuring the SCM CA Vision integration involves:

- Specifying parameters that instruct the broker to start the Synch server at regular intervals.
- Specifying parameters that allow the Synch server to access the CA Vision Web Server.

The Synch Server is a CA Harvest SCM HServer that performs CA Vision synchronizing operations as follows:

1. The Broker starts the Synch Server at the specified interval.
2. The Synch Server obtains new objects from the CA Vision Server and posts any outbound records to the server as required.
3. After the synchronizing operations are complete, the Synch Server process terminates.

Note: The Synch Server is a unique HServer that only performs these CA Vision-related operations. The server does not service regular CA Harvest SCM transactions. Consequently, it does not waste resources for extended periods of time while it is idle.

This section contains the following topics:

Broker and Synch Server Configuration (see page 390)

Broker and Synch Server Configuration

To configure CA Harvest SCM to integrate with CA Vision, add the required parameters to HBroker.arg and Hserver.arg. Then create a caVision.dfo file that contains the login credentials for the CA Vision server.

HBroker.arg

In HBroker.arg, the –cavisionsynchinterval=<nn minutes> parameter specifies how often the Synch Server is activated. This parameter is required to enable the CA Vision integration.

For example, -cavisionsynchinterval=30 specifies the Synch Server is activated after every 30 minutes.

HServer.arg

You can add the following parameters to HServer.arg for the CA Vision integration:

- (Optional) –pvavendpoint=<url>
- (Optional) –pvavproxyhost=<proxy host>
- (Optional) –pvavproxyport=<proxy port number>
- (Optional) –synchlogging=<level>
- (Optional) –cavisionlogdetail=Y

A default Salesforce.com url is imbedded in the synch server. Override the default value using the –pvavendpoint=<url> parameter. You require this value only in special circumstances.

If the CA Harvest SCM Server runs in an intranet using a TCP/IP proxy to access internet URLs, specify the following parameters:

- –pvavproxyhost=<proxy host>
- –pvavproxyport=<proxy port number>

To control logging of the Synch server, use the –synchlogging= parameter. This parameter provides the same logging value as –logging= but applies only to the Synch server. If –synchlogging= is not specified in HServer.arg, the Synch server uses the –logging= value.

To verify that Synch Server details are added to the CAVisionSynch log file, specify -cavisionlogdetail=Y. For details about Synch Server logging, see the Synch Server Logging section.

cavision.dfo

The cavision.dfo file is used to store the CA Vision Server login credentials. Use the CA Harvest SCM command line utility, svrenc, to create this file. The utility stores the user name and password, encrypts in the dfo file. To run svrenc, use the –f option to specify the cavision.dfo file name.

Important! We recommend using a special user id with CA Vision administrator authority who also must be a AgileVision user for the CA Vision Server login used by the sync server. This user must be configured with a Time Zone locale value of GMT+12:00. This configuration ensures that the CA Vision server is accessed without any permission-related issues and that date and time validation is not impacted due to users accessing the system from various demographics.

Example:

```
svrenc  -f cavision.dfo –usr=admin@agile.mycompany.com
-pw=mypwKaHCb0ONhJcOtJ728ULaEyvKW
```

In this command, if you do not specify –usr and/or –pw, svrenc prompts for these parameters.

The password is a combination of the login password and password security token. In the given example, mypw is the password and the remaining characters are the password token. If a token has not been provided, generate it by logging in to CA Vision and clicking Setup. In Setup, expand My Personal Information and select Reset My Security Token. The CA Vision server emails the newly generated token to the specified email address. See the following example for a snippet of the email that is received after resetting the security token.

Example:

Your new security token is as follows. Security tokens are case-sensitive.
User Name: admin@agile.mycompany.com
Security Token: KaHCb0ONhJcOtJ728ULaEyvKW

If the password for the User Name was CaVision#abcd, the full password required for the –pw parameter of svrenc is:

-pw=CaVision#abcdKaHCb0ONhJcOtJ728ULaEyvKW

Email notifications

For certain user-oriented synchronization operations, you can configure the Synch server to email error notifications. For this configuration, specify the same –premail= parameter in HServer.arg as used by the SCM Peer Review feature.

Example:

```
-premail="hsmtp -m mail3.mycompany.com -p 25 -f harvest.user@scmbroker"
```

Synch Server Logging

If logging is enabled via either the –logging= or –synchlogging= parameters, the Synch Server generates a log file named yyyymmddhserversynch.log. This log file contains logging information generated by the Synch Server for that date. Additionally, the Synch Server produces a yyyymmddCAVisionSynch.log file that contains a more user-friendly logging of any errors produced during the synchronizations. If –cavisionlogdetail=Y is specified in HServer.arg, the log also contains details of the CA Vision objects that are retrieved during the Synch Server cycles.

Appendix A: Troubleshooting

Note: The instances of Linux in this section refer to both the Linux and zLinux operating environments.

This section contains the following topics:

How Do I Resolve a "Cannot Start Broker" Error?

Solution:

Check the following items:

- Verify the privileges you have on the computer on which you are trying to start the broker process.
- Check if the process is running.

- The error message could be obtained if somehow the broker process is not able to start to the RTserver process.
- Verify that there is an RTserver process already _started/can_be_started, by this broker process (reference to which RTserver process the broker connects to is given in the HBroker.arg file in the $CA_SCM_HOME and in Windows it is %CA_SCM_HOME% directory).

  Note: The broker ARG file is named hbroker.arg on Windows and HBroker.arg on UNIX and Linux.
- Verify connectivity with the computer hosting the RTserver process.
- Verify if the required port is available for usage or not (default is TCP 5101).

Cannot Start Hserver Error

Symptom:

How do I resolve a "Cannot Start Hserver" error?

Solution:

Check the following items:

- If you are using Oracle, verify that the Oracle TNS Listener is started.
- Check if the maximum number of servers has already been started.
- Make sure the database version matches the server.

Invalid Username/Password/Broker Error

Symptom:

An "Invalid Username/Password/Broker" error appears when I attempt to log in to the client.

Solution:

Check the following items:

- Enter the correct user name and password.
- Use the correct broker in the Broker Name field in the Login dialog.

- Use the same host name as the bkrd process that is pointing to a broker based on UNIX computers.
- Verify the basic network connectivity with the broker computer from the client computer.
- Close any CA Harvest SCM client applications. Identify a file named HClient.cm file on the client computer; try renaming the file or deleting it.

Try to log in again to see if the problem is resolved.

How Do I Resolve a "Connection Timed Out" Error When Connecting to an Agent on a Remote Computer?

Solution:

Check the following items:

- Check if you have the agent up and running.
- Check if the agent is started with an account with root privileges.
- Verify your basic network connectivity: Try to ping the agent from the broker, and try to ping the broker from agent.
- Verify if you are using a correct computer name (host name is different from DNS name).
- Verify if you are using the correct version of the CA Harvest SCM agent and client (includes the product Administrator, Workbench, Web Interface, and command-line utilities).

E3080003: Requested Message Key Not Found Error

Symptom:

When I execute a promote process the following error message appears:

E3080003: Requested Message Key Not Found

Solution:

This message means that you might have more than one setting for CA_SCM_HOME. Check your environmental variable CA_SCM_HOME to make sure that it is defined only once, and for only one installation of the product.

How Do I Resolve the "Error creating DSN harvest: 1 General installer error"?

Valid on Linux

Symptom:

When I install on Linux, run HDBsetup, and use the CO option to create the ODBC DSN, the following message appears:

Error creating DSN harvest: 1 General installer error

Solution:

Remove the following library path statement from the system PATH variables:

```
LD_LIBRARY_PATH = /usr/lib
```

The Broker Process Does Not Start, and Clients Cannot Connect to the Server

Valid on Windows

Symptom:

When the client runs the broker as a service, the broker process does not start and clients cannot connect to the server. If I start a broker from the CA Harvest SCM server, clients can connect.

Solution:

For instructions to start the broker as a service on the server, see Start the Broker as a Service (see page 277).

What ODBC Settings are Required?

Solution:

You can define a default service on any Oracle server, and on an Oracle server you can define several of them. Doing so affords the opportunity for co resident ODBC applications (CA Harvest SCM server) to connect to a default service while bypassing the SQL*NET and TCP/IP layers and gain a significant performance improvement.

On UNIX, the shell environment must have such a matching set of ORACLE_SID and ORACLE_HOME environment variables to exploit a default service and thus a local connection. When the current setting for $ORACLE_HOME (which implies the path to the associated listener.ora file) and $ORACLE_SID match one of listener.ora's SID_DESC blocks, a single "open" default service is defined for that $ORACLE_HOME/$ORACLE_SID pair, and thus for the current process and any child processes it might spawn. If no such matching SID_DESC pair exists, no open default service for the current environment and process exists, and any attempt to make a local connection will fail. For ODBC configuration, in $ODBC_HOME/odbc.ini Oracle datasource should be configured without creating a Servername. This is done by the CAI/PT ODBC installation.

In addition, two settings, Enable Scrollable Cursors and Application Using Threads, should be set to True (value 1 sets to true).

eTPKI Installation Fails

Symptom:

The eTPKI installation fails.

Solution:

Try installing eTPKI manually as follows:

1. Navigate to %CA_SCM_HOME%/install/ETPKI.
2. Run setup.exe, for example:

   ```
   setup.exe caller="SCMClient" install env=all
   ```

Custom Options Are Not Available During Install

Symptom:

The server custom options are not available during installation.

Solution:

Do the following:

1. Insert the installation media into your drive.

   The Product Explorer dialog appears.
2. Right-click the Server component option and select This Feature and all subfeatures will be installed on the local drive.

   The custom options appear.

Fatal Error During Uninstall

Symptom:

A fatal error occurs during the uninstall of a CA Harvest SCM component.

Solution:

This problem occurs when the PATH environment variable size exceeds the limit. Do the following to correct the problem:

1. Record the PATH environment variable setting.
2. Clear the PATH environment variable.
3. Go to the Add/Remove utility and click the support link for the CA Harvest SCM component program.
4. Click Repair.

   The component program is repaired.
5. Click Close.
6. Click Remove.

   The component is uninstalled.
7. Add the PATH environment variable setting.

How Do I Resolve the "Cannot register RTserver" Error?

Solution:

Check the following items:

- The error message could be received if the broker process is unable to start the RTserver process.
- Verify that there is an RTserver process already_started/can_be_started, by the broker process (reference to which RTserver process the broker connects to is given in the HBroker.arg file in the $CA_SCM_HOME and in Windows it is %CA_SCM_HOME% directory).

  Note: The previous broker ARG file is named hbroker.arg on Windows and HBroker.arg on UNIX and Linux.
- Verify connectivity with the computer hosting the RTserver process.
- Verify if the required port is available for usage or not (default is TCP 5101).

If you are using Oracle 9i or 10g and you have Oracle, Tomcat, and the Web Interface installed on the same computer, you may be prompted with an XML database (XDB) logon screen when you attempt to access the Web Interface with a URL.

When your Oracle instance is set up, the XML Database option must not be selected. To fix this problem, consult your Oracle database administrator.

Can I Deploy Software Using an Electronic Software Distribution Tool?

Solution:

Valid on Windows, UNIX, and Linux

Yes, certain predefined lifecycles in CA Harvest SCM help you develop software using the product and deploy the software using a software deployment application. If you have CA Software Delivery (CA Software Delivery) installed, you can use it with the Deploy Release Model lifecycle of the product to develop and deploy software with maximum efficiency.

Note: For more information about these predefined lifecycles and others, see the *Administrator Guide*.

CA Harvest SCM Folder is Not Available in the Folders Explorer or Reports Are Not Listed

Symptom:

The CA Harvest SCM folder is not available in the Folders Explorer or reports are not listed.

Solution:

The reports are stored in a BIAR (Business Intelligence Archive) file. This file is imported during the installation process. If the BIAR file import process failed, you cannot view the reports.

To make the CA Harvest SCM folder available in the Folders Explorer or to list reports, do the following:

1. Import the BIAR file.
2. Configure the database connection.

Follow these steps:

1. Insert the DVD media and select All Programs, BusinessObjects XI, BusinessObjects Enterprise, Import Wizard.

   The Import Wizard appears.

2. Click Next.

   The Source environment screen appears.

3. Select the Business Intelligence Archive Resource (BIAR) option from the Source drop-down list.

4. Navigate to *media drive*/Reports directory and select the file named CA_SCM_R12_EN.biar. Click Next.

   Note: In Windows, *media drive* represents the drive letter; in Linux/UNIX, it represents the mounted directory.

   The destination environment screen appears.

5. Enter the administrator password in the password field.

6. Click Next until the User and groups screen appears. Click Select All, and click Next.

7. Click Next until the Folder and objects screen appears. Click Select All, and click Next.
8. Click Next until the Finish button appears. Click Finish.

   The BIAR file is imported to BusinessObjects Enterprise.
9. Click Done.

   The wizard closes.

Follow these steps:

1. Select All Programs, BusinessObject XI, BusinessObjects Enterprise, Designer.

   The logon screen appears.
2. Enter the Administrator password, and click OK.
3. Select File, Import.

   The Import Universe screen appears and lists universes.
4. Select the CA_SCM_R12_EN universe, and click OK.
5. Select File, Parameters.

   The Universe Parameters dialog appears.
6. Click Edit and enter the database user name and password for the CA Harvest SCM database. Select the DSN which is configured to access the CA Harvest SCM database. If no DSN for CA Harvest SCM exists, create an ODBC DSN. Click Next.
7. Click Test Connection to test the connection for the database.
8. Click Next until the Finish button appears. Click Finish.

   The Universe Parameters dialog appears.
9. Click OK.
10. Save the universe and exit the designer.

    When you open InfoView, you can view and run the reports.

Unable to Run the Reports

Symptom:

I am unable to run the reports.

Solution:

This problem occurs due to improper connection settings.

Follow these steps:

1. Go to All Programs, BusinessObject XI, BusinessObjects Enterprise, Designer.

   The logon screen appears.
2. Enter the Administrator password if any, and click OK.
3. Go to File, Import.

   The Import Universe Screen appears and lists universes.
4. Select CA_SCM_R12_EN universe, and click OK.
5. Go to File, Parameters.

   The Universe Parameters dialog appears.
6. Click Edit.
7. Enter the database user name and password for the CA Harvest SCM database. Select the DSN which is configured to access the CA Harvest SCM database.

   Note: If there is no DSN for CA Harvest SCM, create an ODBC DSN.
8. Click Next. Click Test Connection to test the connection for the database.
9. Continue clicking Next until the Finish button appears. Click Finish.
10. Click OK in the Universe Parameters dialog.
11. Save the universe and exit the designer.

    The connection is set and you are able to run the reports.

BusinessObjects Services Not Running

Symptom:

I am unable to log on to InfoView or when I run a report, an error message appears, for example:

```
Input File Repository Server not responding
```

Solution:

This problem occurs because BusinessObjects services might stop running due to external interruptions. Restarting the particular service might solve the problem.

Follow these steps:

1. Go to All Programs, BusinessObjects XI, BusinessObjects Enterprise, Central Configuration Manager.

   The Central Configuration Manager main screen lists the services available.

2. Right-click the Server which is the cause of the problem, and select Start (or Restart if the service is already running).

Note: In BusinessObjects, a running server has two states: Enabled or Disabled. Even if the server is running if it in disabled state, it will generate exceptions.

Follow these steps:

1. Go to All Programs, BusinessObjects XI, BusinessObjects Enterprise, Central Configuration Manager.

   The Central Configuration Manager window appears.

2. Select the Enable/Disable Servers from the toolbar.

   A log on window appears.

3. Enter the user name, password, and other credentials, and click Connect.

   A dialog lists servers with corresponding states.

4. To enable a server, check the box beside it. To disable a server, clear the check box.

5. Click OK to save the changes.

Crystal Reports Inconsistency in Data Displayed for Prompt

Valid on Crystal Reports

Symptom:

When I schedule a report, data displayed in the parameter list is inconsistent. The parameter list shows the following inconsistencies:

- Data that is not relevant to the CA Harvest SCM instance
- Data that is not updated (old data)

Solution:

Crystal Reports that are run against the universe in BusinessObjects XI has a known issue where the scheduling parameters are not refreshed.

Note: BusinessObjects provides technical details about the behavior of parameter lists in the document *cr_xi_refreshing_universe_LOV.pdf* at their website: http://support.businessobjects.com.

The current workaround for this issue is not a complete solution; it involves the following basic steps:

1. Manually refresh the report using Central Management Console to resolve the parameter list refresh problem. You do this every time want to refresh the list.
2. Add a registry key to update values in the list for on-demand reports. You do this only once.

Follow these steps:

1. Click Start, Programs, Business Objects XI, Business Object Enterprise, and select Business Object Enterprise Java Administration Launchpad. Click the Central Management Console link.

   The Central Management Console appears.
2. Click Folders from the Organize section.
3. Click CA Reports from the available folders.
4. Click CA Harvest SCM from the available list, and click the Objects tab.

   The Objects tab lists all the reports.
5. Click a Crystal Report, for example, Item Summary by Project. (Crystal Reports are denoted by a green diamond icon.)

   The report properties display in a window.
6. Click the Refresh Options link below the Properties tab.

7. Click Select All, Refresh Report, and Update.

   Crystal Reports you selected are refreshed and up to date.

8. Follow Steps 5 through 7 to refresh other Crystal Reports.

Follow these steps:

1. Open the Registry Editor using the regedit command.
2. Create a subkey named Database under HKEY_LOCAL_MACHINE\SOFTWARE\Business Objects\Suite 11.0\Crystal Reports\.
3. Create a String value under Database key, name it AlwaysRefreshUniverseLOV, and enter the value as '1'.

   The registry key is added.

Index

O

P

R

S

U

W

Z