HP Select Identity Software

Connector for Solaris BSH UCA

Connector Version: 1.00

Installation and Configuration Guide

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

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

Support

You can visit the HP software support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

To find more information about access levels and HP Passport, go to:

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

Contents

1 Documentation Map

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

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

Figure 1 Documentation Map

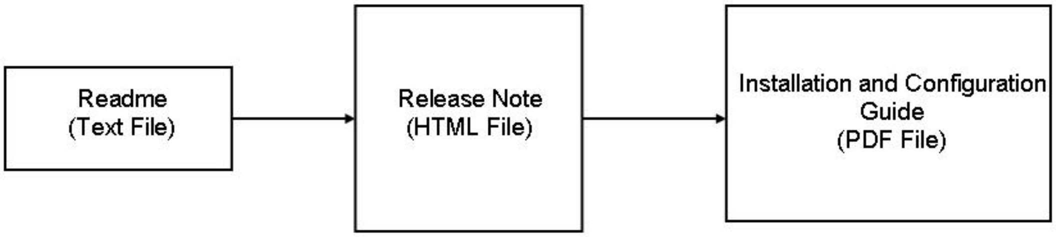


Table 1 **Connector Documentation**

Document Title and Filename	Contents	Location
Release Note `Solaris_BSH_UCA Connector 1.00 Release Note.htm`	This file contains necessary information on new connector features, enhancements, known problems or limitations, and support information.	`/Docs/` subdirectory under the connector directory
Connector Installation and Configuration Guide `Solaris_BSH_UCA_guide.pdf`	The connector installation guide provides installation instructions for a specific connector. It contains resource-specific configuration details. The connector installation guide also provides detailed information on: • Deploying a connector on an application server. • Configuring a connector with Select Identity. Refer to this guide when you need information about connector installation.	`/Docs/` subdirectory under the connector directory

2 Introduction

This chapter provides an overview of the HP Select Identity connector for Solaris BSH UCA. An HP Select Identity connector for Solaris BSH UCA enables you to provision users and manage identities on a UNIX application server configured for mutual authentication.

About this Guide

The *HP Select Identity Solaris BSH UCA Connector Installation and Configuration Guide* provides an overview of how to install, configure, and deploy the Solaris BSH UCA connector on the Select Identity server.

The instructions explained in this guide are unique for the Solaris BSH UCA connector. For additional-connector specific or resource-specific installation information, refer to the specific connector's installation and configuration guide.

About HP Select Identity

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

Select Identity supports both one-way secure socket layer (SSL) authentication in which only the server is authenticated and two-way (mutual) SSL authentication in which both the server and client are authenticated.

When implementing mutual authentication and using a UCA-based connector, the application server and clients are configured and set up with the appropriate initialized keystores and truststores.

About Connectors

Select Identity connectors are either agent-based or agent-less. Agent-based connectors communicate with an agent module installed on the resource platform to do all forward provisioning. The agent also communicates with Select Identity web service over HTTP with SPML payloads.

You can establish a connection between a resource and Select Identity by using a connector. A connector is a piece of code that resides in the same system as the Select Identity core product in order to communicate with target systems such as a directory server or database. A connector is resource specific. The combination of Select Identity and connector enables you to perform a set of tasks on the resource to manage identity. A connector can be *unidirectional* or *bidirectional*. A unidirectional connector helps you manage identities from Select Identity, but if any change takes place in the resource, the resource cannot communicate that change back to Select Identity. On the other hand, a bidirectional connector can reflect the changes made on the resource back to Select Identity. This property of bidirectional connectors is known as *reverse synchronization*.

Features and Capabilities

A connector enables Select Identity to access a resource to manage users, groups, and entitlements. Select Identity can typically perform the following tasks by using a connector:

- Add, update, and remove users
- Retrieve user attributes
- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords

The set of tasks that can be performed by the connector on a resource varies from connector to connector.

A connector usually consists of:

- **RAR** file— this compressed file contains the connector binaries.
- **Schema** file — this contains the mapping file for the connector. A mapping file contains resource attribute information of the connector, which must be linked to Select Identity attributes.

A connector may not contain a schema file. In that case, the mapping file for the connector can be generated by using the Select Identity attribute mapping utility. Refer to *Appendix C: Attribute Mapping* chapter of *HP Select Identity Administration Guide* for information on the attribute mapping utility.

In addition to the above two files, there could be other files packaged with the connector, such as an agent file, a script file, and so on.

Deploying a Connector

To use a connector with Select Identity, you must deploy it on an application server, and then configure it with Select Identity. The RAR file of the connector, which contains the binary files, must be deployed on an application server. You must perform the following tasks to deploy and configure a connector. These tasks are explained throughout this guide.

1. Extracting Contents of the Schema File
2. Installing the Solaris BSH UCA Connector
3. Installing the UCA Agent
4. Deploying the Solaris BSH UCA Connector on the Application Server
5. Configuring the Solaris BSH UCA Connector with Select Identity

About Unified Connector Architecture

A connector implemented with Unified Connector Architecture (hereafter referred to as UCA) satisfies the requirements for secure communication between a connector and agent and communication between an agent and Select Identity's web service. To ensure a high level of security, UCA uses client authentication, also referred to as mutual authentication.

Mutual authentication requires each client and server (communication endpoints) to present a valid certificate, issued for the purpose of either a client or the server authentication respectively, that is trusted by the other endpoint and has not yet been revoked.

➤ Mutual authentication does not require the client to use the keystore and truststore. These are Java concepts. However, to set up mutual authentication, it may be necessary to create a keystore and a truststore on both ends before setting up a UCA connector and agent. Refer to the *HP Select Identity Installation Guide*.

UCA provides a generic framework to make connector development consistent and easy, as well as highly secure.

Implementation of UCA:

- Eliminates redundant connector coding efforts
- Simplifies deployment by conforming features and functions across connectors
- Enhances operating security
- Supports pluggability and scalability
- Simplifies administration
- Enables automatic connector updates
- Supports management standards such as WS/MAN

Benefits of UCA are:

- Developer codes the resource-specific portions of a connector; the generic UCA framework handles the common functions such as fail-over, polling, security, and logging.
- Agent-less and agent-based connectors look and behave similarly.
- Supports JCA 1.0 and 1.5.

- Agent-based communication channels and protocols are 'standardized' and 'unified'. This allows loose coupling and simplifies administration and security.
- Live update service enables remote agent-code upgrades.
- All agent configuration occurs centrally.
- Enables uniform/standards-based management and deployment.

➤ Some of the above features may not be available in the current version of this connector.

About the Solaris BSH UCA Connector

The Solaris BSH UCA connector enables HP Select Identity to manage user data on UNIX systems. The Solaris BSH UCA connector is a one-way connector and pushes changes made to user data in the Select Identity database to the target Solaris server. The Solaris BSH UCA connector is an agent-based connector that utilizes mutual authentication for enhanced security. Mapping files included with the connector control how Select Identity maps fields within Select Identity to Solaris BSH fields.

During mutual authentication, Select Identity can serve as either the client for outbound communication, or the server for inbound communication. When a Select Identity connector communicates with a Solaris BSH UCA connector agent, the agent is the server and Select Identity and the Solaris BSH UCA connector are the client.

When mutual authentication is implemented, you must associate the certificate with the connector if a web service request is made with the connector's identity. Connectors use the certificate to authenticate themselves to Select Identity during a web service call.

The Solaris BSH UCA connector enables HP Select Identity to perform the following tasks on the Solaris server:

- Add, update, and remove users
- Enable and disable users
- Verify a user’s existence
- Change user passwords
- Reset user passwords
- Retrieve all entitlements
- Retrieve a list of supported user attributes
- Grant and revoke entitlements to and from users

⚠ Do not use the user’s primary group as an entitlement. If the primary group is changed, it remains as an entitlement in Select Identity but not on the resource.

The Solaris BSH UCA connector is a unidirectional connector and pushes changes made to user data in the Select Identity database to a target server. The mapping file controls how Select Identity fields are mapped to `/etc/passwd` fields.

This connector can be used with Select Identity version 4.20.

Overview of Installation Tasks

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

Table 2 Organization of Tasks

Task Number	Task Name	Reference
1	Install Select Identity (if not already installed)	See the *HP Select Identity Installation Guide*.
2	Install the Solaris BSH UCA connector agent. Includes: • Editing the property files to point to the new keystore and truststore. • Modifying the server.xml file to point to the new server keystore and truststore.	See Installing the UCA Agent on page 19.
3	Install the Solaris BSH UCA connector on the Select Identity server.	See Installing the Solaris BSH UCA Connector on page 17.
	• Meet the system requirements for the connector.	See System Requirements on page 17.
	• Extract the contents of the Schema file (the file that contains the mapping files for the connector) to the location on the Select Identity server.	See Extracting Contents of the Schema File on page 18.
4	Set up the keystore and truststore if not already set up. • Create keys. • Copy the files to the Select Identity server and the Domino connector server directory.	See Set Up the Keystore and Truststore on page 19, and refer to the *HP Select Identity Installation Guide.*
5	Deploy the connector on application server.	See Deploying the Solaris BSH UCA Connector on the Application Server on page 31.
6	Configure the connector in the Select Identity application.	See Configuring the Solaris BSH UCA Connector with Select Identity on page 33.
7	Deploy the resource and create the service in the Select Identity application.	See Configuring the Solaris BSH UCA Connector with Select Identity on page 33.

3 Extracting the Contents of the Schema File

Most of the connectors contain at least one schema file. This file contains the mapping information of the connector. Some of the connectors do not have a schema file packaged with them, and the mapping files are generated with the help of the Select Identity attribute mapping utility. If the connector that you are deploying does not contain a schema file packaged with it, skip to the next chapter.

You must extract contents of the schema file to a location on the Select Identity server. Perform one of the procedures explained below depending on the application server on which the connector will be deployed.

WebLogic

1 Create a subdirectory in the Select Identity home directory where you can store the connector mapping files.

  For example, you can create `<SI_HOME_DIR>/Schema` where

  `<SI_HOME_DIR> = /opt/si420/weblogic/` for Select Identity installed on UNIX and `<SI_HOME_DIR> = C:\si420\weblogic\` for Select Identity installed on Windows.

2 Extract contents of the schema JAR file, `unixBshSchema.jar,` to the `Schema` directory. Some connectors may contain more than one schema file. Refer to the connector's installation and configuration guide to find out the right schema file to be used.

3 To ensure that the `CLASSPATH` environment variable in the WebLogic startup script references the `Schema` directory created above, perform the following steps:

  a Open the `myStartWL.cmd`/`.sh` file from the location `<SI_HOME_DIR>/weblogic/scripts` with a text editor.

  b Add the directory path of the `Schema` directory to the `CLASSPATH` variable in the script.

  If you install more than one connector, you can extract the `Schema` file of all the connectors to the same location.

WebSphere

On WebSphere, `<WebSphere_Install_Directory>/AppServer/lib/ext` folder is present in `WAS CLASSPATH` by default. Extract contents of the schema file to the location `<WebSphere_Install_Directory>/AppServer/lib/ext`.

4 Installing the Solaris BSH UCA Connector

The Solaris BSH UCA connector is packaged in the following files, which are located in the `Solaris - BSH - UCA` directory of the Select Identity Connector CD.

Table 3 Solaris BSH UCA Connector Files

Number	File Name	Description
1	`UCAUnixConnector_420.rar`	Contains the WebSphere binaries for the connector
	`UCAUnixConnector_420WL9.rar`	Contains the WebLogic v9.2 binaries for the connector
2	`unixBshSchema.jar`	Contains the mapping file: • `UnixConnector.xml`

System Requirements

The Solaris BSH UCA connector is supported in the following environments:

Table 4 Platform Matrix for Solaris BSH UCA connector

Select Identity Version	Application Server	Database
4.20	WebLogic 9.2 on Windows 2003	Microsoft SQL Server 2000
	WebSphere 6.1 on HP-UX 11i	Oracle 9i

The UCA agent communicates with UNIX by using Secure Shell (SSH).

When using SSH method, you must ensure that the SSH daemon is running on the Solaris server to which you will provision users by using the connector.

The Solaris BSH UCA connector and UCA agent are supported on Solaris 8 and 9.

The agent must be installed on the system where the Solaris 8 and 9 server is running.

The Solaris BSH UCA connector and agent work *only* with Select Identity version 4.20.

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

- The Select Identity server should be configured for internationalization. Refer to the *HP Select Identity Installation Guide* for details.
- The resource must be configured to support local language characters.

Extracting Contents of the Schema File

The schema file of the connector contains necessary mapping information to map resource attributes to Select Identity. Extract contents of the `unixBshSchema.jar` file to a directory that is in the application server `CLASSPATH`.

Installing the Connector RAR

To install the connector RAR file (`UCAUnixConnector_420.rar` or `UCAUnixConnector_420WL9.rar`) on the Select Identity server, you must copy the file to a local subdirectory on the Select Identity server, and then deploy on the application server.

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

5 Installing the UCA Agent

The Solaris BSH UCA connector is an agent-based connector. The UCA Agent is a suite of services and support files deployed on the resource.

Perform the following tasks to complete the installation process:

1. Set Up the Keystore and Truststore — Perform this task to create the keystore and truststore that you will need when you install the UCA Agent.
2. Install the UCA Agent — Perform this task to install the UCA Agent by using the installation wizard.
3. Start the UCA Agent — Perform this task to start the UCA Agent on Windows and UNIX.

Set Up the Keystore and Truststore

This keystore is used to store the mutual authentication key pair. You register this keystore in Select Identity using the Security Set Up feature. Refer to the *HP* Select Identity *Administration Online Help*.

The Solaris BSH UCA connector supports mutual authentication and requires a mutual authentication key pair to be stored in the keystore. When you install the UCA Agent, you are prompted for the name and location of your keystore and truststore where these key pairs are stored.

➤ It is not necessary to create a new keystore and truststore specifically for use with this connector. The Solaris BSH UCA connector and agent can work with any keystore and truststore that you already have, as long as you add the mutual authentication key pair as required.

If the keystore and truststore have not been previously created, create the keystore and truststore before installing the agent.

To create a mutual authentication keystore for use in Select Identity 4.20, perform the following steps:

1. Run the keytool utility to create a keystore and a key pair.

   ➤ When you create a key pair, a keystore is automatically created during this process.

2. Generate a certificate request file, as shown in this command line example which creates an X.509 certificate request file at `./req/myReq.csr` for a certificate at `myKeyAlias` in the keystore:

   ```
   keytool -certreq -keyalg RSA -alias myKeyAlias -file ./req/myReq.csr
   -keystore ./ks/myKeystore -storetype JKS
   ```

3. Send the new request file to your certificate authority for digital signing.

4 Import the signed certificate back to the keystore from which you generated the certificate request. The following command line example imports the signed certificate file `./signed/signedCert.pem` to `ks/myKeystore` at the key alias named `myKeyAlias`:

```
keytool -import -trustcacerts -alias myKeyAlias -file ./signed/
signedCert.pem -keystore ./ks/myKeystore -storetype JKS
```

5 Import the signed certificate to the appropriate truststore. The following command line example imports the signed certificate file `./signed/signedCert.pem` to `ks/mytruststore` at the key alias named `myKeyAlias`:

```
keytool -import -trustcacerts -alias myKeyAlias -file ./signed/
signedCert.pem -keystore ./ks/mytruststore -storetype JKS
```

6 Generate the property files for the keystore and/or truststore by executing either `genprop.sh` (Unix) or `genprop.bat` (Windows).

When prompted to specify the file type to generate, select the appropriate option:

- For keystores, select option **2:OVSI secure object migration keystore**
- For truststores, select option **3:OVSI truststore**

Install the UCA Agent

Complete the following steps to run the installation wizard, which installs the agent. You can also install the agent as a console application or as a service.

Before running the installation wizard, ensure that the `log4j-1.2.8.jar` file resides in the `JREDIR\lib\ext` directory, where *JREDIR* is the Java Runtime Environment (JRE) that will be used by the wizard.

For example, if the JRE resides in

`C:\Program Files\Java\j2re1.5.0_12`,

verify that the

`log4j-1.2.8.jar` file

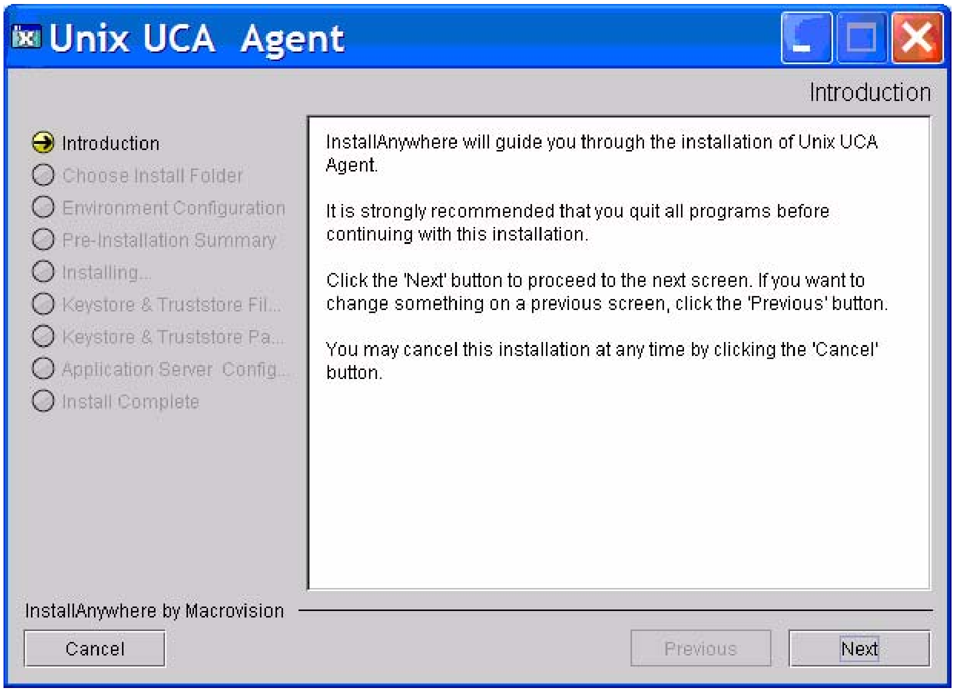
resides in

`C:\Program Files\Java\j2re1.5.0_12\lib\ext`.

Also, ensure that this JRE is included in the `path` system variable.

Note: JRE 1.5 is required.

1 Run the installation wizard (`install.exe`). The Introduction page opens:


Unix UCA Agent
Introduction
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore Fil...
Keystore & Truststore Pa...
Application Server Config...
Install Complete
InstallAnywhere by Macrovision
Cancel
Previous
Next

InstallAnywhere will guide you through the installation of Unix UCA Agent.

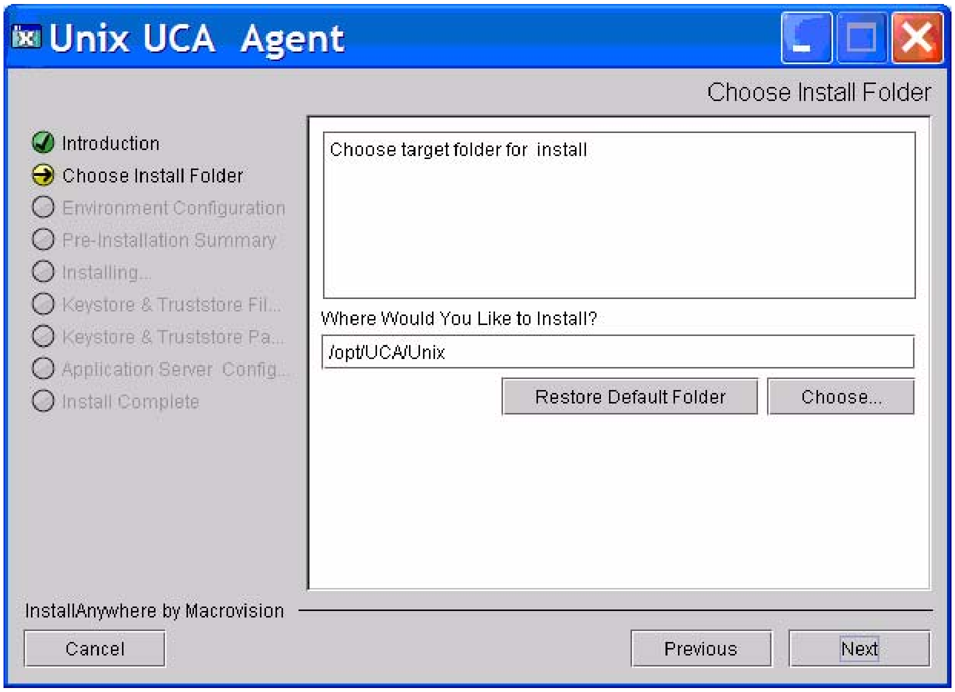
It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

2 Click **Next** to proceed.

The Choose Install Folder page opens.



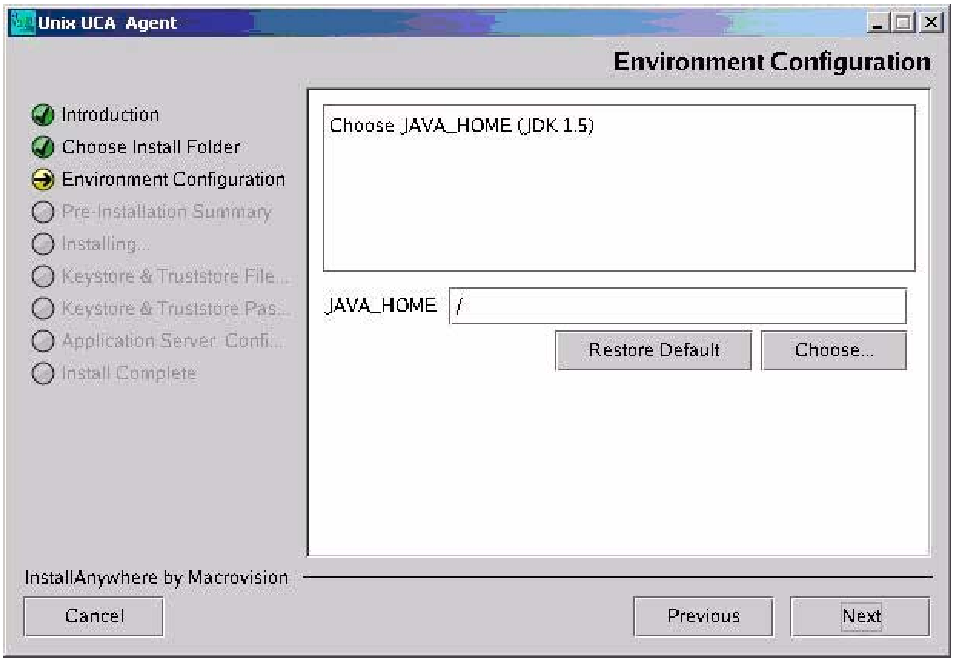
Choose target folder for install

Where Would You Like to Install?

/opt/UCA/Unix

3 Enter your install path or click **Choose** and select a directory location from the list. Click **Next.**

The Environment Configuration page opens.


Unix UCA Agent
Environment Configuration
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore File...
Keystore & Truststore Pas...
Application Server Confi...
Install Complete
InstallAnywhere by Macrovision
Cancel
Previous
Next

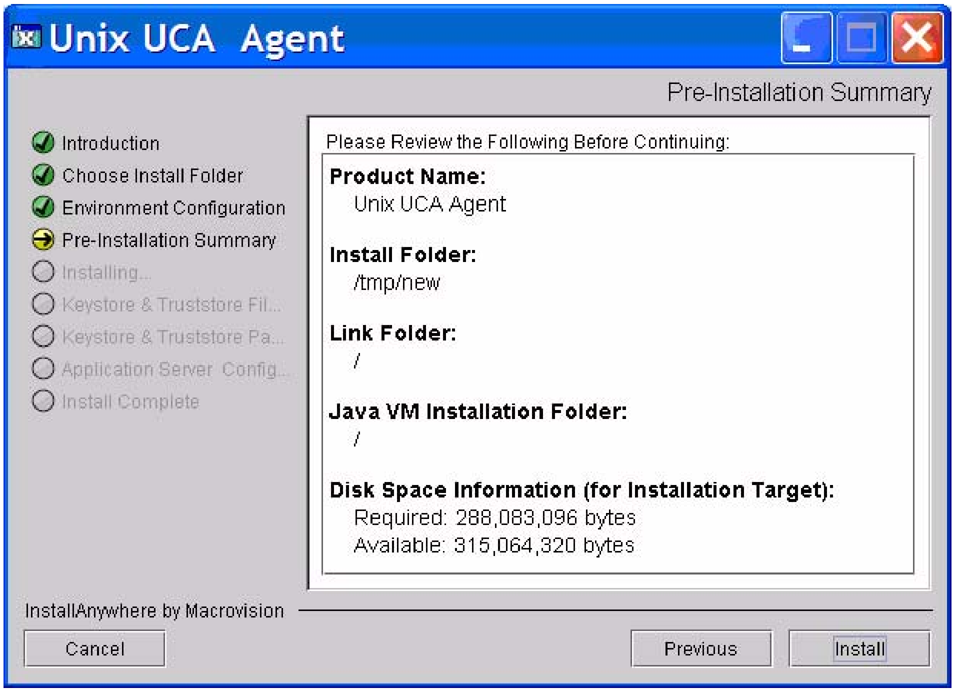
Choose JAVA_HOME (JDK 1.5)

JAVA_HOME /

Restore Default Choose...

4 Select the directory location for **JAVA_HOME** and click **Next**.

The Pre-Installation Summary page opens.



Please Review the Following Before Continuing:

Product Name:
Unix UCA Agent

Install Folder:
/tmp/new

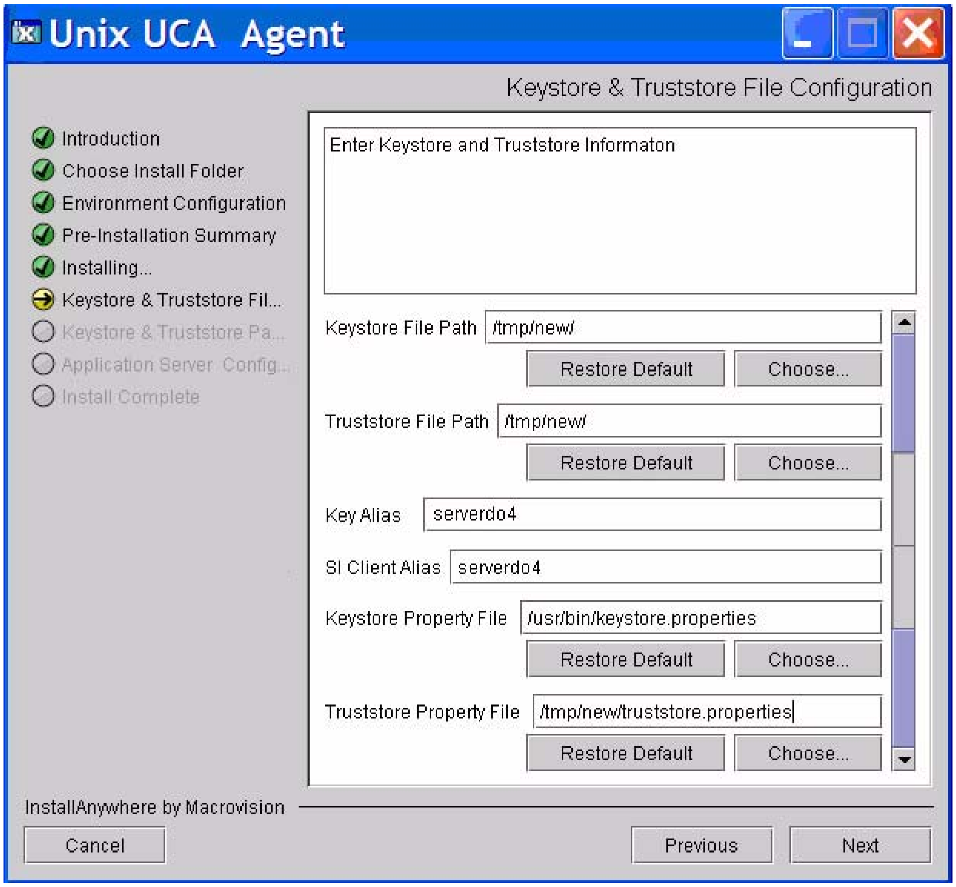
Link Folder:
/

Java VM Installation Folder:
/

Disk Space Information (for Installation Target):
Required: 288,083,096 bytes
Available: 315,064,320 bytes

5 Click **Install** to start the installation.

The Keystore & Truststore File Configuration page opens.


Unix UCA Agent
Keystore & Truststore File Configuration
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore Fil...
Keystore & Truststore Pa...
Application Server Config...
Install Complete
Restore Default
Choose...
InstallAnywhere by Macrovision
Cancel
Previous
Next

Enter Keystore and Truststore Informaton

Keystore File Path: /tmp/new/

Truststore File Path: /tmp/new/

Key Alias: serverdo4

SI Client Alias: serverdo4

Keystore Property File: /usr/bin/keystore.properties

Truststore Property File: /tmp/new/truststore.properties

6 In the **Keystore File Path** field, enter the keystore file name and path or click **Choose** to select the path from the directory list.

7 In the **Truststore File Path** field, enter the truststore file name and path or click **Choose** to select the path from the directory list.

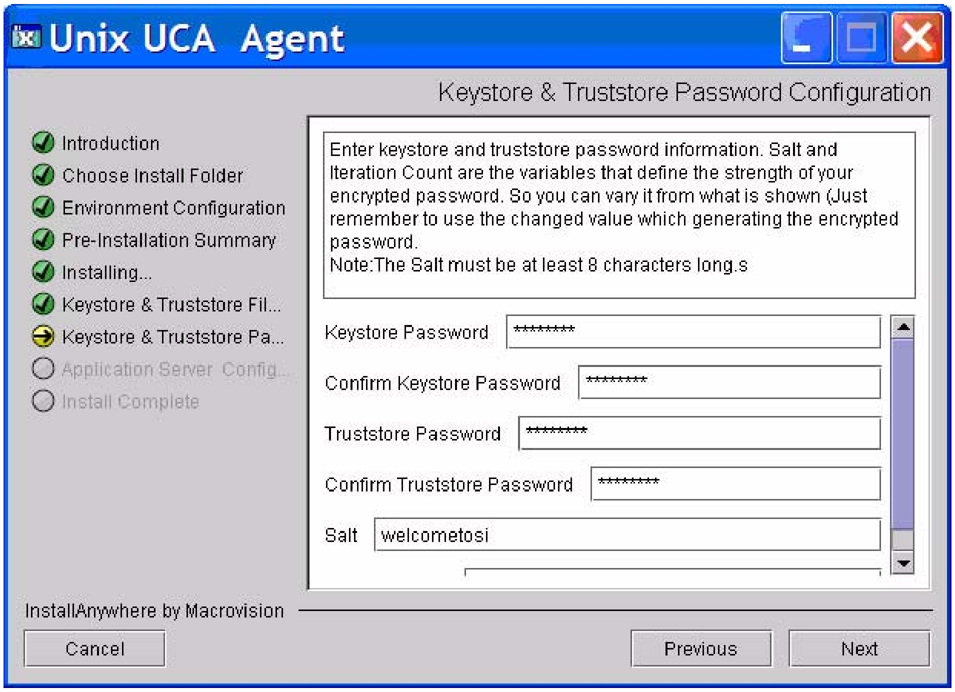
8 In the **Key Alias** field, enter the key alias name.

9 In the **SI Client Alias** field, enter the SI Client Alias name.

10 In the **Keystore Property File** field, enter the keystore property file name and location or click **Choose** and select a directory location from the list.

11 In the **Truststore Property File** field, enter the truststore property file name and location or click **Choose** and select a directory location from the list. Click **Next**.

12 Click **Next**. The Keystore & Truststore Password Configuration page opens.


Unix UCA Agent
Keystore & Truststore Password Configuration
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore Fil...
Keystore & Truststore Pa...
Application Server Config...
Install Complete
InstallAnywhere by Macrovision
Cancel
Previous
Next

Enter keystore and truststore password information. Salt and Iteration Count are the variables that define the strength of your encrypted password. So you can vary it from what is shown (Just remember to use the changed value which generating the encrypted password.
Note:The Salt must be at least 8 characters long.s

Keystore Password ********
Confirm Keystore Password ********
Truststore Password ********
Confirm Truststore Password ********
Salt welcometosi

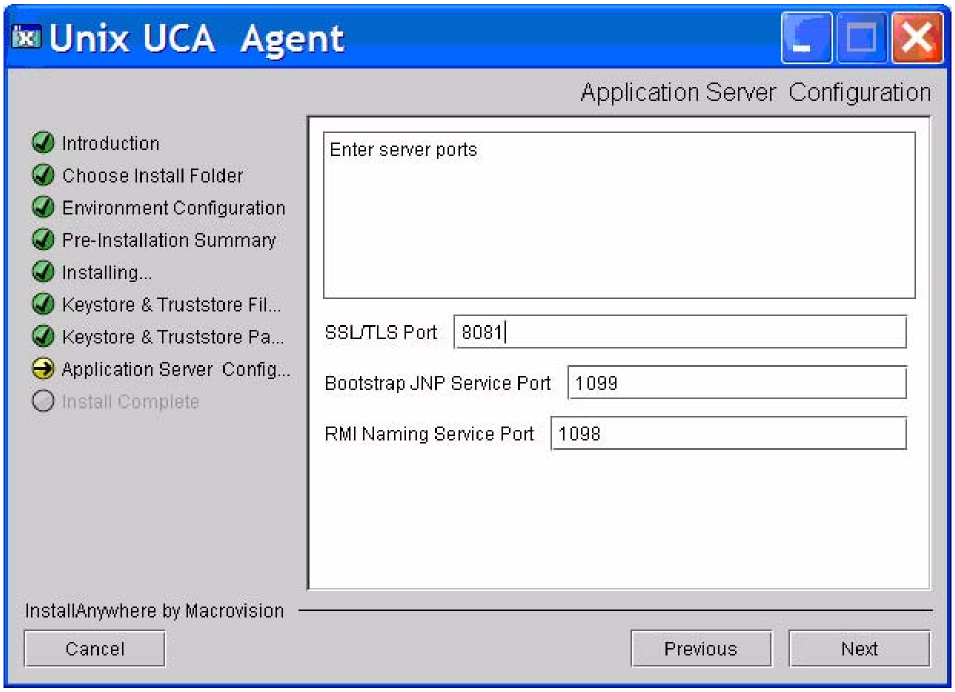
13 Enter the keystore and truststore passwords.

The keystore and truststore passwords are those that you created before beginning this installation procedure.

14 In the **Salt** field, enter the salt value for encryption. Salt must be at least 8 characters long.

15 Scroll down to view the **Iteration Count** field, and enter the iteration count.

16 Click **Next**. The Application Server Configuration page opens.



Unix UCA Agent
Application Server Configuration
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore Fil...
Keystore & Truststore Pa...
Application Server Config...
Install Complete
InstallAnywhere by Macrovision
Cancel
Previous
Next

Enter server ports

SSL/TLS Port: 8081

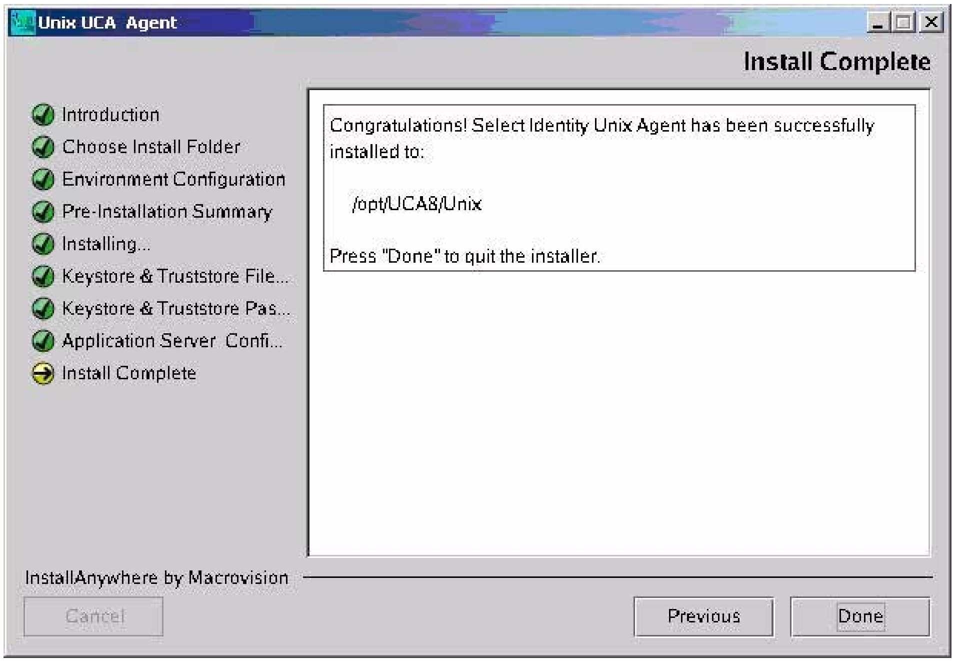
Bootstrap JNP Service Port: 1099

RMI Naming Service Port: 1098

17 Use the default server ports, or enter your server ports, if provided.

> The SSL/TLS port is the port used on the Select Identity server side to communicate with the agent.

18 Click **Next**. The Install Complete page opens.



Unix UCA Agent
Install Complete
Introduction
Choose Install Folder
Environment Configuration
Pre-Installation Summary
Installing...
Keystore & Truststore File...
Keystore & Truststore Pas...
Application Server Confi...
Install Complete
InstallAnywhere by Macrovision
Cancel
Previous
Done

Congratulations! Select Identity Unix Agent has been successfully installed to:

/opt/UCA8/Unix

Press "Done" to quit the installer.

19 Click **Done** to exit the install wizard.

Install the UCA Agent Manually

It may be necessary to make changes to the UCA Agent configuration after installation. Use the manual procedure described below to implement changes. The following procedure explains how to install the UCA Agent manually.

The following files are created or modified:

${**Install directory**}\server\minimal\deploy\jbossweb-tomcat55.sar\server.xml
${**Install directory**}\server\minimal\deploy\security-service.xml (new file)
${**Install directory**}\server\minimal\deploy\jbossweb-tomcat55.sar\META-INF\jboss-service.xml

To manually configure the UCA Agent or make changes to the existing configuration, complete the following steps as appropriate:

1 Edit the connector configuration in `server.xml` as follows:

   a Remove the following attributes:

   - `keystoreFile`
   - `keystorePass`
   - `truststoreFile`
   - `truststorePass`

   b Add the following attributes:

   - `securityDomain`
   - `SSLImplementation`

   Your script should look like the following example:

```
<Connector port="8081" address="${jboss.bind.address}"
  maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
  emptySessionPath="true"
  scheme="https"
  secure="true"
  clientAuth="want"
  keyAlias="server"
  sslProtocol = "TLS"
  securityDomain="java:/jaas/encrypt-keystore-password"
  SSLImplementation="org.jboss.net.ssl.JBossImplementation" />
```

The port is the SSL/TLS port.

2. Create a new file named `security-service.xml` in the following directory:

   `${`**Install directory**`}\server\minimal\deploy`

   This file should contain the following code:

```
<server>
 <!-- ============================================================== -->
 <!-- Security                                                       -->
 <!-- ============================================================== -->
 <mbean code="com.hp.si.uca.jaasSecurity.UcaJaasSecurityDomain"
name="jboss.security:service=PBESecurityDomain">
      <constructor><arg type="java.lang.String"
value="encrypt-keystore-password"/></constructor>
      <attribute name="KeyStoreURL">${jboss.server.home.dir}/keystores/
serverdb</attribute>
      <attribute name="KeyStorePass">

{CLASS}org.jboss.security.plugins.FilePassword:${jboss.server.home.dir}/
keystores/keystore.password
      </attribute>
      <attribute name="TrustStoreURL">${jboss.server.home.dir}/keystores/
servertrustdb</attribute>
      <attribute name="TrustStorePass">

{CLASS}org.jboss.security.plugins.FilePassword:${jboss.server.home.dir}/
keystores/truststore.password
      </attribute>
      <attribute name="Salt">welcometosi</attribute>
      <attribute name="IterationCount">13</attribute>
   </mbean>
</server>
```

   In the above script:

   - `KeyStoreURL` and `TrustStoreURL` specify the paths of the keystore and truststore.
   - `keystore.password` and `truststore.password` are the files that contain the encrypted passwords that you create in the next step.
   - `Salt` and `IterationCount` are the variables that define the strength of the encrypted password. Use the same values for `Salt` and `IterationCount` in the next step. The value of `Salt` must be at least 8 characters long.

3. Edit `jboss-service.xml` and add the following line at the end:

```
<depends>jboss.security:service=PBESecurityDomain</depends>
```

4. Move the `jboss-service.xml` file to the following directory:

   `${`**Install directory**`}\server\minimal\keystores`

5. Run the following commands to generate `keystore.password` and `truststore.password`:

```
java -cp ../lib/jbosssx.jar org.jboss.security.plugins.FilePassword
welcometosi 13 password keystore.password

java -cp ../lib/jbosssx.jar org.jboss.security.plugins.FilePassword
welcometosi 13 password truststore.password
```

➤ Confirm java -version shows 1.5 or plus.

In the above commands:

— **welcometosi** and **13** are the values of `Salt` and `IterationCount` used in `security-service.xml`. These values should be the same for both `keystore.password` and `truststore.password`.

— **password** is the password of the keystore/truststore being encrypted.

Start the UCA Agent

To start the UCA Agent, go to `INSTALL_PATH\bin` and invoke `run.sh`.

Set up the Solaris BSH UCA Connector to Work with Select Identity

To set up Select Identity to work with the Solaris BSH UCA connector, complete the following steps in the Select Identity user interface:

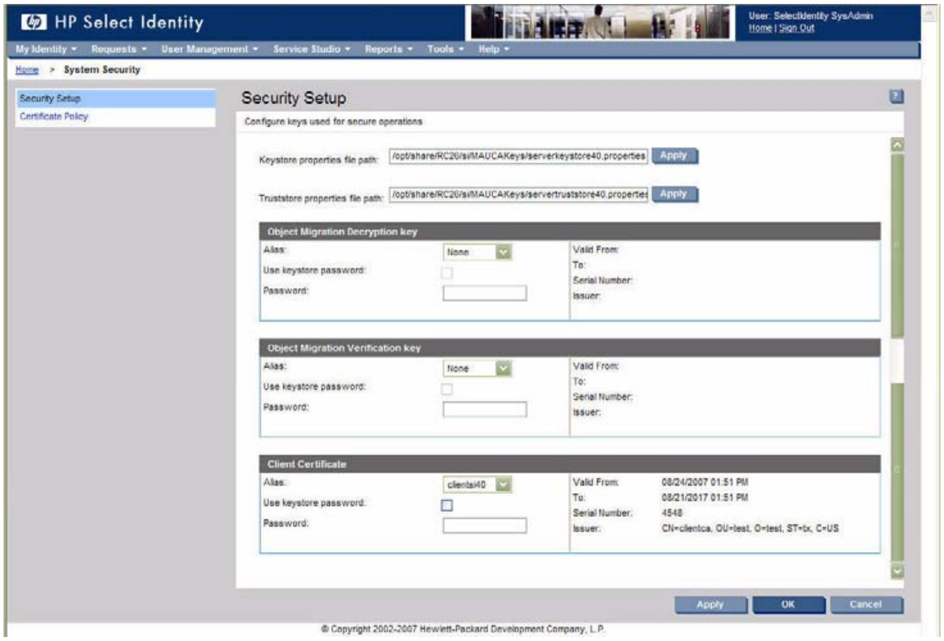
1 Generate the properties files.

For example, `clientkeystore.properties` and `clienttruststore.properties`.

For `ucaSiClientDb` and `ucaSiClientTrustDb`, do the following:

2 Under **Tools** menu, select **System Security** → **Security Setup**.

The Security Setup page opens.



Security Setup

Configure keys used for secure operations

Keystore properties file path: /opt/share/RC20/si/MAUCAKeys/serverkeystore40.properties

Truststore properties file path: /opt/share/RC20/si/MAUCAKeys/servertruststore40.propertie

Object Migration Decryption key

Alias:	None	Valid From:	
Use keystore password:	□	To:	
Password:		Serial Number:	
		Issuer:	

Object Migration Verification key

Alias:	None	Valid From:	
Use keystore password:	□	To:	
Password:		Serial Number:	
		Issuer:	

Client Certificate

Alias:	clientsi40	Valid From:	08/24/2007 01:51 PM
Use keystore password:	□	To:	08/21/2017 01:51 PM
Password:		Serial Number:	4548
		Issuer:	CN=clientca, OU=test, O=test, ST=tx, C=US

© Copyright 2002-2007 Hewlett-Packard Development Company, L.P.

3 Enter the file path for the keystore properties file.

4 Enter the file path for the trust store properties file.

5 Under the **Client Certificate** section select the client certificate alias from the list. For example, *clientsi40*.

Refer to the *HP Select Identity Installation Guide* and the *HP Select Identity Administration Online Help* for information about setting up security features in Select Identity and implementing mutual authentication.

6 Deploying the Solaris BSH UCA Connector on the Application Server

To install the connector on Select Identity, you must first deploy the connector on the application server.

WebLogic/WebSphere

To deploy the connector on a WebLogic or WebSphere application server, perform the following steps:

1. Create a subdirectory in Select Identity home directory where you can store the connector's RAR file.

   For example, you can create *<SI_HOME_DIR>*`/connectors` where *<SI_HOME_DIR>* = `/opt/Select_Identity` in UNIX and *<SI_HOME_DIR>* = `C:\Select_Identity` in Windows. (A connector subdirectory may already exist.)

2. Copy the RAR file from the Select Identity Connector CD to the connector subdirectory.

3. Perform the following steps to deploy the connector on WebLogic. If deploying on WebSphere, skip to step 4 on page 31.

   a. If not currently running, start the application server in the domain for Select Identity and log on to the WebLogic Server Console.

   b. In the left panel, expand the Deployments folder, and then right-click on **Connector Modules**, and select **Deploy a New Connector Module**.

      Alternatively, at the right panel of the Server Console home page, click the **Connector Modules** link, which is under Your Deployed Resources column of the Domain Configurations section. The Resource Connectors page appears. Click on **Deploy a New Connector Module** link on this page.

   c. Click the link in the Location field, locate, and select the RAR file from the list. The RAR file is stored in the connector subdirectory.

   d. Click **Target Module**.

   e. If only one server is configured, skip to the next step. If more than one server is configured, the next page prompts you to select the servers on which you want to deploy the connector. Select the server instance, and then click **Continue**.

   f. Review the settings. Keep all the default settings and click **Deploy**. The Status of the Last Action column should display Success.

4. If you want to deploy the connector on WebSphere, perform the following steps:

   a. Start the application server, if necessary.

   b. Log on to the WebSphere Application Server Console.

   c. Navigate to **Resources** → **Resource Adapters**.

d Click **Install RAR**. The Install RAR File page appears.

e If it is a cluster setup, select a WebSphere node from the Node list.

f In the Server path field, enter the path to the connector's RAR file. The RAR file is stored in the subdirectory created in the beginning.

g Click **Next**.

h In the Name field, enter a name for the connector.

i Click **OK**.

j Click the **Save** link (at the top of the page).

k On the Save to Master Configurations dialog box, click **Save**.

l Repeat step f to step k for all the available nodes (for cluster setup).

m Click **Resources** → **Resource Adapters**. The Resource Adapters page appears.

n Click **Browse Nodes** and select a node in the Node text box (in case of a cluster) and click **Apply**.

o Click the new connector.

p Click **J2C Connection Factories** in the Additional Properties table.

q Click **New**.

r In the Name field, enter the name of the factory for the connector. This is the pool name of the connector. Refer to the respective connector's installation and configuration guide to find out the specific pool name.

s Click **OK**.

t Click the **Save** link.

u On the Save to Master Configurations dialog box, click **Save**.

v Repeat the step m to step u for all available nodes (for cluster setup).

w Restart WebSphere.

7 Configuring the Solaris BSH UCA Connector with Select Identity

This chapter describes the procedure to configure the Solaris BSH UCA connector with Select Identity and the connector-specific parameters that you must provide when configuring the connector with Select Identity.

Configuration Procedure

After you deploy the connector RAR on the application server, you must configure the connector with Select Identity. Perform the following tasks to deploy and configure the Solaris BSH UCA connector with Select Identity.

1. Add a New Connector
2. Add a Resource

Add a New Connector

The first task in deploying the Solaris BSH UCA connector to work with Select Identity is to add the new connector in Select Identity by using Select Identity's user interface. Use one connector for each supported server resource type. Multiple resources can use the same connector. For example, to connect to three servers, you install and deploy only one connector. Resources sharing a connector are placed in a resource pool.

Connector management defines the communication criteria Select Identity uses to reconcile identity information with your system resources. Before a connector can interface between Select Identity and the designated resources, there must be a record of the connector in Select Identity.

Connector records cannot be created unless the connector has already been deployed on the WebLogic or WebSphere console server. To add the connector using the Select Identity user interface, perform the following steps:

Refer to the following table when entering the parameters in the Manage Connectors fields.

Table 5 Manage Connectors Page

Field	Description
Connector Name	Identifies the complete name of the connector.
Pool Name	Specifies the full name of the resource pool for this resource. If the pool name is incorrect, you cannot add the connector. For example, `eis/UCAUnixConnector`.
Mapper Available	Indicates whether the Attribute Mapper utility supports the connector. Select **No** for the Attribute Mapper utility support option when setting up the Solaris BSH UCA connector.
Approval Required	Indicates approval is required for changes when configuration change control is in place. Approval Required is the default setting.

Refer to the *HP Select Identity Administration Online Help* for information on adding a new connector in Select Identity.

Add a Resource

After you add the connector to Select Identity, you set up a resource. Add a new resource in Select Identity that uses the newly added Solaris BSH UCA connector. Use the Select Identity Add Resource wizard to perform the procedure. Steps 1-3 are explained in this section as they relate to the UCA connector configuration and setup.

Refer to the *HP Select Identity Administration Online Help* for complete instructions on adding a resource in Select Identity.

Add Basic Information

To deploy a resource that uses the newly added connector, perform the following steps:

1. Select **Service Studio > Resources**.
2. Click **Add New Resource**.
3. Complete the fields as required.

Refer to the following table when entering the parameters in the Add Resource: Basic Information page:

Table 6 Add Resource: Basic Information page

Field	Description
Resource Description	Provides a brief description of the resource.
Connector Name	Identifies the connector used to access the resource in Select Identity. **Note:** The connector must be included in the Managed Connectors list for you to select it here. If you do not see your connector, map the necessary connector before continuing. See the *HP Select Identity Administration Online Help* for more information.
Authoritative	Indicates whether the resource is authoritative. The system default is No, or non-authoritative. Select Identity defines two classes of resources: • An Authoritative resource is considered to be the “master” source for important user identity accounts and attribute values. • Non-authoritative resources typically need to stay synchronized with regard to the key aspects of a user’s identity that originate from the authoritative resource and also contain other resource-specific identity data. For more information about authoritative and non-authoritative resources, refer to the *HP Select Identity Administration Guide*.
Select Identity Password Authority	Indicates the authority level of the password. The default setting is No, but change this setting to Yes if both of the following are true: • The password on this resource authenticates users logging into Select Identity. • Password changes made through the "Forget Password" function are updated synchronously on the resource. **Note:** Selecting Yes allows login data to be synchronized across all resources so that users only need to sign in once.
Delete User	Indicates the conditions during which users are deleted from the resource. The default setting is No.
Approval Required	Indicates approval is required for changes when configuration change control is in place. See the *HP Select Identity Administration Online Help* for information. Approval Required is the default setting.
Resource Owner	Indicates the name (User ID) of the resource owner if a contact person has been assigned to answer questions about this resource.

Create a Mutual Authentication Policy

The Add Resource: Mutual Authentication Policy page enables you to create a mutual authentication policy by specifying the inbound and outbound security settings. The inbound security settings apply to incoming web service requests. The outbound security settings apply to outgoing connections to the connectors.

➤ Configuration of mutual authentication is required for the Solaris BSH UCA connector.

Complete the fields as required.

- Select the inbound security levels.
- Select the outbound security levels.

  ➤ You *must* select “Server and Client Certificate Required” as the outbound security level when configuring mutual authentication for the Solaris BSH UCA connector.

Refer to the *HP Select Identity Administration Online Help* for instructions.

Refer to the following table when entering the parameters in the Add Resource: Create a Mutual Authentication Policy page:

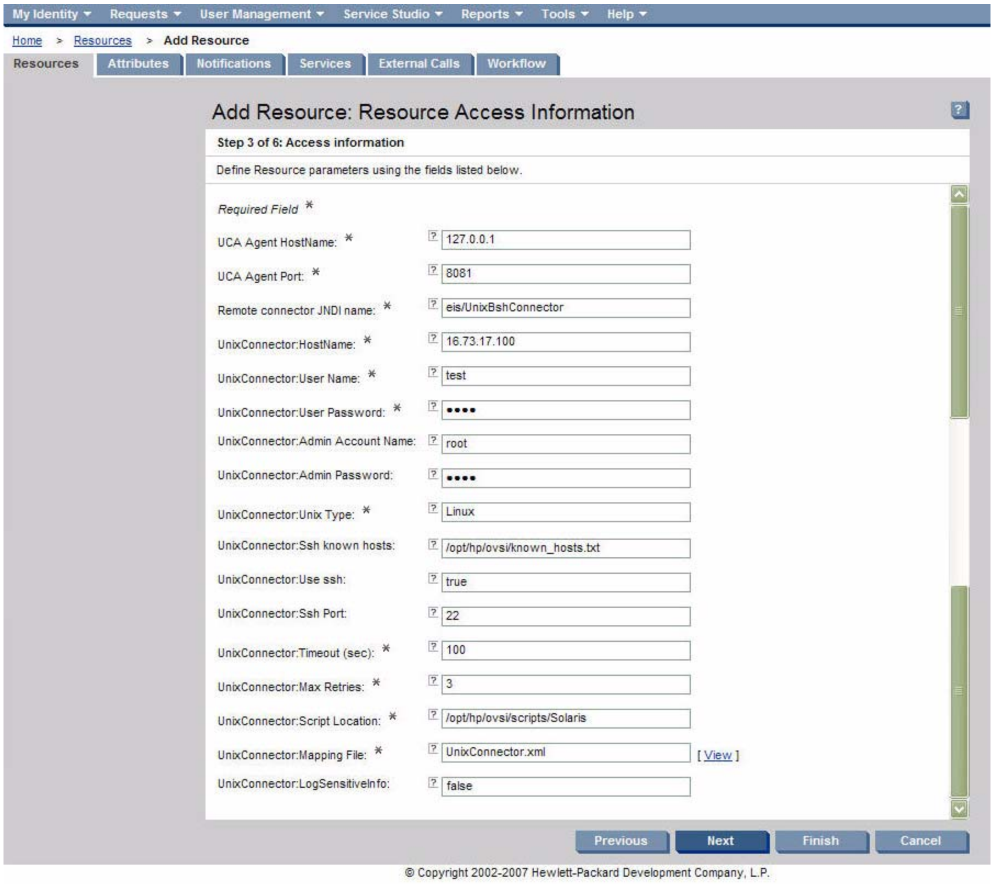
Table 7 Add Resource: Mutual Authentication Policy page fields

Field	Description
Inbound Communication – Security Level	Indicates the inbound security level selected from the following Security Level options: • **None** – indicates that the resource does not use PKI (Public Key Infrastructure) for secure inbound communication. If Client Certificate Required is selected on the System Security page, you cannot select None. A client certificate is required. • **Client Certificate Required** – indicates that the client must present a certificate when connecting to Select Identity.
Inbound Communication – Only Allow Resource Owner Submit Request	If checked, indicates that the Resource Owner must be defined and the resource owner must have a certificate. When selected, only the owner of the resource can submit a reconciliation request.
Outbound Communication – Security Level	Indicates the outbound security level selected from the following Security Level options: • **None** – indicates that the resource does not use PKI for secure outbound communication. • **Server Certificate Required** – indicates that the server must present a certificate when Select Identity connects to this server. The Select Identity connector must also request the server's certificate and validate it. • **Server and Client Certificate Required** – indicates that the Select Identity connectors must submit a request for the server certificate and validate it, and that the Select Identity connectors must present a certificate to the server for authentication. — **Use SI Certificate** – Indicates that the Select Identity certificate is required and is set up. When you select this option, the following information displays about the SI certificate: issuer, valid from and to date, and serial number. If you select **Use SI Certificate**, no further selections are required. **Note:** If you do not select the Use SI Certificate option, you must complete the following fields: — **Client Certificate** – Select an available client certificates from the drop-down list. — **Use keystore password** – Indicates that the password for the keystore is used. — **Password** – Specifies a password to use if not using the keystore password.

Set up Resource Access Information

Use the Resource Access Information page to define access values for the new resource. The fields that display on the Add Resource: Resource Access Information page are based on your previous entries.

The Add Resource: Resource Access Information page resembles the following example:



Add Resource: Resource Access Information

Step 3 of 6: Access information

Define Resource parameters using the fields listed below.

Required Field *

UCA Agent HostName: *	127.0.0.1
UCA Agent Port: *	8081
Remote connector JNDI name: *	eis/UnixBshConnector
UnixConnector:HostName: *	16.73.17.100
UnixConnector:User Name: *	test
UnixConnector:User Password: *	••••
UnixConnector:Admin Account Name:	root
UnixConnector:Admin Password:	••••
UnixConnector:Unix Type: *	Linux
UnixConnector:Ssh known hosts:	/opt/hp/ovsi/known_hosts.txt
UnixConnector:Use ssh:	true
UnixConnector:Ssh Port:	22
UnixConnector:Timeout (sec): *	100
UnixConnector:Max Retries: *	3
UnixConnector:Script Location: *	/opt/hp/ovsi/scripts/Solaris
UnixConnector:Mapping File: *	UnixConnector.xml [View]
UnixConnector:LogSensitiveInfo:	false

© Copyright 2002-2007 Hewlett-Packard Development Company, L.P.

1 Complete the fields as required.

Refer to Table 8 when entering the parameters in the Add Resource: Resource Access Information page:.

Table 8 Add Resource: Resource Access Information page

Field	Sample	Description
UCA Agent HostName	127.0.0.1	Host on which UCA agent is running.
UCA Agent Port	8081	UCA Agent SSL/TLS port.
Remote connector JNDI name	eis/UnixBsh Connector	JNDI name of connector factory on agent side.
UnixConnector:Host Name	16.73.17.100	Host name IP address of the4 UNIX server.
UnixConnector:User Name	test	Name of the Solaris user.

Table 8 Add Resource: Resource Access Information page (cont'd)

Field	Sample	Description
UnixConnector:User Password	**********	The password of the Solaris user in Solaris.
UnixConnector:Admin Account Name	root	Login account (a super user) with the privilege to provision other users on this system. The default value is root.
UnixConnector:Admin Password	**********	Password of the super user that has the privilege to provision users on this system.
UnixConnector:Unix Type	Solaris	The type of the UNIX operating system. This can only be one of the following: "Linux" or "Solaris" or "HPUX" or "AIX".
UnixConnector:Ssh known hosts	/opt/hp/ovsi/ known_hosts.txt	Indicates the name of a text file on the application server that contains the list of hosts and their host keys for the SSH protocol. Needed only for SSH communication.
UnixConnector:Use ssh	true	A **True** value indicates to use SSH when connecting to the resource. **Note:** You must select True for the Solaris BSH UCA connector.
UnixConnector:Ssh Port	22	If you use SSH, the default is 22; otherwise, the default is 23.
UnixConnector:Timeout (sec)	100	Number of seconds to wait for a request to complete.
UnixConnector:Max Retries	3	Number of times the request should be retried upon a connection timeout given in the above parameter.
UnixConnector:Script Location	opt/ovsi/scripts/	The directory location of the BSH scripts that are used by the connector. These are the scripts extracted from `unix-bsh-scripts.zip`. This is the directory on the Solaris server.
UnixConnector:Mapping File	UnixConnector. xml	Name of the connector mapping file used to map resource attributes to Select Identity attributes.
UnixConnector:LogSensitiveInfo	False	Control the logging of sensitive information. Options are **True** or **False**. The default is False.

2 Click **Finish** to exit the Add Resource wizard or click **Next** to proceed to the next step.

After completion of step 7, you can save the resource without entering values on the remaining pages. You can modify the resource later to add the mapping, reconciliation, and caching information. Refer to the *Select Identity Administration Online Help* for information about creating and modifying resources.

Map Resource Attributes

After successfully adding a resource for the Solaris BSH UCA connector, you must map the resource attributes to Select Identity attributes. Refer to the *HP Select Identity Administration Online Help* for information on mapping and creating attributes.

When you first access the Add Service: Map Resource Attributes page, some of the Select Identity attributes (column two) may be pre-populated with values if the resource attribute (column one) exactly matches a Select Identity attribute. For the unmapped attributes indicated by "(Select one)", map these attributes as appropriate.

Map the resource attributes as required by performing the following steps:

1. Under Resource Attribute, review the list of resource attributes.
2. From the Attribute list, choose the Select Identity attribute (column two) to map to the resource attribute (column one). For information about attributes, see Attributes.
3. Determine how the attributes are updated during reconciliation by choosing one or both of the following options:
   - **Sync In** – changes to the resource attribute update the corresponding attribute in Select Identity.
   - **Sync Out** – changes to the Select Identity attribute update the corresponding resource attribute.

   Resource attributes must be set to "Sync In" to create updates in Select Identity during reconciliation. Resource attributes updated by Select Identity during reconciliation are set to "Sync Out" to push the changes to the resource.
4. Repeat steps 2 and 3, as required, to map all resource attributes.
5. Click **Next** to continue or **Finish** to exit the wizard.

When mapping the attributes, refer to the following table for resource specific mapping information.

Table 9 Solaris BSH UCA Mapping Information

Select Identity Resource Attribute	Connector Attribute	Attribute on Solaris Resource	Description
UserName	username	login (login argument)	UNIX logon name.
Password	password	password	Logon password
Directory	directory	login home directory	User's home directory
Shell	shell	login shell	UNIX logon shell
PrimaryGroup	defaultgroup	login primary group	Default group membership
Description	description	login comment	Comment section in / etc/passwd.

After mapping the attributes, you can use the connector to create a service, or you can associate the connector with an existing service. Refer to the *HP Select Identity Administration Guide* for information on Select Identity services.

Define a Caching Policy (Add Resource Wizard)

The Add Resource: Define Caching Policy page enables you to create entitlement caching to reduce the impact to system performance caused by retrieving service provisions (entitlements) from resources. If you do not enable caching, all entitlements are retrieved and updated through the connector during the reconciliation, service creation, and user ID creation processes. All of the throughput can place significant demands on the system and cause substantial performance degradation.

To define a caching policy, perform the following steps:

1. Select the **Caching Enabled** option to turn caching on.
2. Complete the fields, as appropriate.

Refer to the following table when entering the remaining parameters for the Add Resource: Define a Caching Policy page:

Table 10 Add Resource: Define a Caching Policy page:

Field	Description
Never Expires	Identifies if and how the policy will expire. • Days – number of days between expiration interval • Hours – number of hours between expiration interval • Minutes – number of minutes between expiration interval
Polling Enabled	Indicates that Select Identity should poll this resource. When polling is enabled, Select Identity runs a batch job to poll the resource for changes to entitlements. This synchronizes Select Identity and resource entitlements.
Polling Interval	Indicates the polling interval by specifying how often Select Identity polls the resource. **Note:** This field is only available if you enable polling. • Days – number of days between each polling event • Hours – number of hours between each polling event • Minutes – number of minutes between each polling event
Refresh Cache Now	Specifies a manual refresh operation. Select Identity retrieves entitlements from the resource and updates the entitlement cache. If a large number of entitlements exists, you may have to wait for the retrieval to complete.

Configuring the Solaris BSH UCA Connector on Non-English Platforms

If you install the connector, which is internationalized, on a non-English platform, you will have the following limitations while configuring the connector:

- When entering user attributes to provision (in Select Identity), you cannot enter local language characters for the following attributes:
  - UserName
  - Password
  - Email
- The attribute names on the resource cannot contain non-English characters. Thus, you cannot include non-English characters in the mapping file.
- Non-English entitlements are not supported by the connector.
- All configuration and property file names must be in English.

- The exception messages from the resource are in English.
- The log messages are in English.

8 Uninstalling the Solaris BSH UCA Connector

If you want to uninstall a connector from Select Identity, perform the following steps:

- Remove all resource dependencies.
- Delete the connector from Select Identity.
- Delete the connector from the application server.

See the *HP Select Identity Administration Online Help* for information about deleting a connector from Select Identity and the application server.

Deleting the Connector from Select Identity

Before deleting a connector, remove all dependencies on the connector.

To delete a connector, perform the following steps:

1. Select **Service Studio** → **Resources.**

   The Resource list opens.

2. Click **Manage Connectors**.

   The Manage Connectors page opens.

3. Select the connector.
4. Click **Delete**.
5. Click **OK** to confirm and delete the connector from the list.
6. Click **OK** to return to the **Resource** list.

   ➤ Approval may be required for the deletion, if configuration management is enabled for connectors.

Deleting the Connector from WebLogic

Perform the following to delete a connector from WebLogic:

1. Log on to the WebLogic Server Console.
2. Expand the Deployments folder on the left pane, and then double click **Connector Modules**

   Alternatively, at the right panel of the Server Console home page, click **Connector Modules** link, which is under the **Your Deployed Resources** column in the Domain Configurations section.
3. The right hand panel of the console displays a table showing all the deployed connectors. Click the delete icon next to the connector that you want to uninstall.
4. Click **Yes** to confirm the deletion.
5. Click **Continue**.

Deleting the Connector from WebSphere

Perform the following steps to uninstall the connector from WebSphere:

1. Log on to the WebSphere Application Server Console.
2. Navigate to **Resources** → **Resource Adapters**.
3. Select the connector to uninstall.
4. Click **Delete**.
5. Click the **Save** link (at the top of the page).
6. On the Save to Master Configuration dialog box, click the **Save** button.
7. If it is a cluster setup, click **Browse Nodes** to select other available nodes and perform step 3 to step 6 for each node.

Uninstalling the UCA Agent

To uninstall the UCA Agent, perform the following steps:

1. Stop the agent, if running.
2. Run the following:

   $INSTALL_DIR$\ Uninstall_UCA_Unix_Agent\ Uninstall UCA Unix Agent.exe
3. Click **Next** each time you are prompted.
4. Click **Finish** when the procedure is complete.

A Understanding the Scripts

This connector uses a Java-based scripting engine called BeanShell (**http://www.beanshell.org**). The BeanShell scripts control the interactions between the UCA agent and the UNIX resource. Each is described in this appendix.

The bean shell scripts control how the interaction with the UNIX system is made. You can customize the scripts if you want to customize the interaction for provisioning. The `common.inc` script has all the common variables and methods. Other individual scripts have operation-specific variables and methods.

Table 11 Scripts in `unix-bsh-scripts.zip`

Script Name	Description	Parameters Passed
`common.inc`	Commonly used variables and methods. Defines the UNIX prompt that the connector expects and the UNIX commands to be used for provisioning	
`dotest.bsh`	Test the initial connectivity to Solaris, including login as regular user, super user and check for the existence of /etc/passwd file	`login, loginPassword, rootPassword, connection_timeout`
`finduser.bsh`	Checks for existence of user in Solaris	`username, connection_timeout`
`adduser.bsh`	Adds a new user to Solaris by preparing all the options needed for the UNIX command	`args` (semi-colon-separated name=value pairs), `username, connection_timeout`
`modifyuser.bsh`	Modify the comment, shell, primary or secondary groups of the user	`args` (semi-colon-separated name=value pairs), `username, connection_timeout`
`expirepasswd.bsh`	Locks the password entry for the user	`username, connection_timeout`
`changepasswd.bsh`	Changes the user's password	`username, password, connection_timeout`

Table 11 Scripts in `unix-bsh-scripts.zip` **(cont'd)**

Script Name	Description	Parameters Passed
`changestatus.bsh`	Change the status of the user	`args` (semi-colon-separated name=value pairs), `username`, `connection_timeout`
`deleteuser.bsh`	Delete the user	`username`, `connection_timeout`
`findgroup.bsh`	Get details of a UNIX group	`group`, `connection_timeout`
`listgroups.bsh`	Get all UNIX groups, Get groups of a specific user, Get users of a specific group	`connection_timeout`, `username`, `connection_timeout`, `gname`, `connection_timeout`
`listusers.bsh`	Get all users in system	`connection_timeout`

B Troubleshooting

The information in this appendix will help you diagnose configuration problems that you may encounter when using the connector to provision users on the resource.

User creation request fails with the following errors:

- `Failed to execute operation, Response=UX: useradd: ERROR: /bin/ssh is not a valid shell. Choose another.`

  Possible cause:

  Invalid Shell value for the user.

  Solution:

  Verify the existence of the `/bin/` subdirectory on the host system.

- `Failed to execute operation, Response=UX: useradd: ERROR: group mail1 does not exist. Choose another.`

  Possible Cause:

  Invalid group value given for the primary group

  Solution:

  See the available groups in the `file /etc/group` on the host machine and provide a value from the available list of groups.

Resource creation request fails with the following error:

- `inline evaluation of: sessionLogin(); : Method Invocation session.sendExpect`

  Cause:

  It is caused by unmatched `initialPrompt` of Unix server. You can find default prompt definition in script `common.inc`. The default definition is shown as below, which follows regular expression syntax:

  ```
  initialPrompt = ".*(%|#|\\$|>|:).*$";
  ```

  Solution:

  Check the prompt after user login and prompt after root user login. If the prompt does not match the `initialPrompt`, change the `initialPrompt` in `common.inc` to make it match the prompt of Unix server.

  If the login banner contains some characters that are matched by the regular expression in `initialPrompt`, the connector might also fail. So, edit the `initialPrompt` to match the actual prompt exactly or as close as possible so that it does not match the login banner. For example, If there is a “`:`” or a “`$`” sign in your login banner the `initialPrompt` will match it and will pre-maturely assume it is the actual prompt.