HP Service Activator

Installation Guide

Edition: V62-1A

for the Red Hat Enterprise Linux 6.4 operating system

Manufacturing Part Number : None
October 14, 2013

Legal Notices

Warranty.

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices.

Trademark Notices.

Java™ is a registered trademark of Oracle and/or its affiliates.

Linux is a U.S. registered trademark of Linus Torvalds.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

Red Hat® Enterprise Linux® is a registered trademark of Red Hat, Inc.

EnterpriseDB® is a registered trademark of EnterpriseDB.

Postgres Plus® Advanced Server is a registered trademark of EnterpriseDB.

Oracle® is a registered trademark of Oracle and/or its affiliates.

UNIX® is a registered trademark of the Open Group.

Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Document id: p158-pd001109

Contents

Contents

In This Guide

This guide describes the preinstallation requirements and provides the installation instructions for HP Service Activator.

Audience

The audience for this guide is the Systems Integrator (SI). The SI has some or all of the following background:

- Understands and has a solid working knowledge of
  - — UNIX® commands
- Understands networking concepts and language
- Understands database programming and management
- Is able to program in Java™ and XML
- Understands security issues

Conventions

The following typographical conventions are used in this guide.

Font	What the Font Represents	Example
Italic	Book or manual titles, and manpage names	Refer to the *HP Service Activator — Workflows and the Micro-Workflow Manager* and the *Javadocs* manpage for more information.
	Provides emphasis	You *must* follow these steps.
	Specifies a variable that you must supply when entering a command	Run the command: `InventoryBuilder` <*sourceFiles*>
	Parameters to a method	The *assigned_criteria* parameter returns an ACSE response.
Bold	New terms	The **distinguishing attribute** of this class...
`Computer`	Text and items on the computer screen	The system replies: `Press Enter`
	Command names	Use the `InventoryBuilder` command ...
	Method names	The `get_all_replies()` method does the following...
	File and directory names	Edit the file `$ACTIVATOR_ETC/config/mwfm.xml`
	Process names	Check to see if `mwfm` is running.
	Window/dialog box names	In the `Test and Track` dialog...
	XML tag references	Use the `<DBTable>` tag to...
`Computer Bold`	Text that you must type	At the prompt, type: **`ls -l`**
Keycap	Keyboard keys	Press **Return**.
`[Button]`	Buttons on the user interface	Click `[Delete]`. Click the `[Apply]` button.

Font	What the Font Represents	Example
`Menu Items`	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select `Locate:Objects->by Comment`

Install Location Descriptors

The following names are used throughout this guide to define install locations.

Descriptor	What the Descriptor Represents
$ACTIVATOR *$ACTIVATOR_OPT*	The base install location of Service Activator. The UNIX location is `/opt/OV/ServiceActivator` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\`
$ACTIVATOR_ETC	The install location of specific Service Activator files. The UNIX location is `/etc/opt/OV/ServiceActivator` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\etc\`
$ACTIVATOR_VAR	The install location of specific Service Activator files. The UNIX location is `/var/opt/OV/ServiceActivator` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\var\`
$ACTIVATOR_BIN	The install location of specific Service Activator files. The UNIX location is `/opt/OV/ServiceActivator/bin` The Windows location is `<install drive>:\HP\OpenView\ServiceActivator\bin\`

$JBOSS_HOME	The install location for JBoss.
	The UNIX location is
	`/opt/HP/jboss`
	The Windows location is
	`<install drive>:\HP\jboss`

$JBOSS_DEPLOY	The install location of the Service Activator J2EE components.
	The UNIX location is
	`/opt/HP/jboss/standalone/deployments`
	The Windows location is
	`<install drive>:\HP\jboss\standalone\deployments`

$JBOSS_EAR_LIB	The location for libraries (Java *.jar files) to be executed by the HPSA engine (workflow manager and resource manager). The UNIX location is `/opt/HP/jboss/standalone/deployments/hpsa.ear/lib` The Windows location is `<install drive>:\HP\jboss\standalone\deployments`\hpsa.ear\lib
$ACTIVATOR_DB_USER	The database user name you define. Suggestion: `ovactivator`
$ACTIVATOR_SSH_USER	The Secure Shell user name you define. Suggestion: `ovactusr`

1 Preparing to Install the Product

This chapter provides an overview of the hardware and software requirements for the installation of HP Service Activator. When your site meets all of the requirements described in this chapter, proceed to the instructions in “Installing the Product” on page 31 to complete your Service Activator installation.

Upgrading Service Activator

If you are upgrading from an older version of Service Activator, do not follow the instructions provided in this document. Instead, please refer to the *HP Service Activator Migration Guide*.

Getting Started

Service Activator requires two types of configurations: the **Service Activator server** and the **target machines**[1]. The Service Activator server can be installed on an HP-UX, Windows Server 2008 R2, or Linux operating system and it requires a database server.

This manual describes the installation of Service Activator on the Linux Service Activator server. The server must be configured with each of the preinstallation packages referenced in this chapter, as well as the Service Activator components discussed in Chapter 2, "Installing the Product," on page 31.

This manual also describes the installation and configuration of software required for target machines.

Understanding Hardware Requirements

The Service Activator server system must meet the following, minimum requirements:

- x86-64 platform
- Red Hat Enterprice Linux 6.4
- 2 GB memory
- Disk space available as follows:
  - 1 GB in the /opt partition
  - 1 GB under /etc
  - 1 GB under /var
- The database system requires room for an Oracle® or Postgres Plus® Advanced Server database instance of at least 1 GB for Service Activator data.

NOTE The disk space requirements listed here are minimal requirements for AutoPass and Service Activator. Additional disk space may be required for Oracle, the Java JDK, and Secure Shell. To determine minimum disk space requirements for each of these applications, please consult the pertinent product literature.

Target machines are not limited to any specific type of hardware. They could be computers (running HP-UX, Windows 2000, Windows 2003, Windows 2008, Solaris, or Red Hat Linux operating systems), routers, HLRs, Ethernet switches, or other telecommunications equipment.

1. The term "target machine" refers to the targets of plug-ins, for example such pieces of equipment as a switch, HLR, UNIX machine, Windows machine, etc.

Understanding Software Requirements for the Service Activator Server

Install and configure the following software on an Service Activator server, in the order listed, prior to installing Service Activator:

1. Red Hat Enterprise Linux 6.3 and all available patches. It is crucial that all available patches are installed for the operationg system. In particular, you must install the recommended Java patches that are available at the Java download site.
2. The ksh shell and X11 must be installed. If they are not installed as part the initial installation they can be installed by running the following two commands as root:
   a. yum install ksh
   b. yum groupinstall “X Window System”
3. Java SE update 37 JDK or later (version 6, but not version 7).
4. Oracle 11g or PPAS 9.2. The database does not need to be installed on the Service Activator server; it can be installed on any server that is accessible to the Service Activator server
5. Microsoft Internet Explorer 9.0, Firefox 24, or Chrome 30.

Instructions for installing each of these base products are provided beginning on page 18.

NOTE The Service Activator Installation includes the JBoss application server used by Service Activator. The version of JBoss needed by Service Activator is included on your installation CD and will be installed automatically in the appropriate location when Service Activator is installed. You do not need to install JBoss separately. Use of any other version of JBoss is not supported.

Understanding Software Requirements for Target Machines

The requirements for the target machines depends on the specific plug-in that is used. Study the documentation for the plug-ins for more information.

If the built-in mechanisms for script deployment and execution are used, the target machines must run an SSH server.

Addressing Localization Requirements

If Service Activator is deployed in a non-English environment, all Service Activator components including JBoss, Oracle, PPAS, and the Secure Shell server must be running under the same locale.

To avoid encoding mismatches between the Service Activator server and the targets, targets should use the same encoding as the Service Activator server. For example, if an HP-UX Service Activator server is running under Japanese locale (ja_JP.SJIS), a Solaris target should do the same using ja_JP.PCK.

Installing the Base Products

Use the following information and instructions to install the required software on your Service Activator server.

Installing Java SE update 37 JDK

Go to `http:http://www.oracle.com/technetwork/java`, and download the Java SE for Linux x64 self-extracting binary file. You can choose the default options shown.

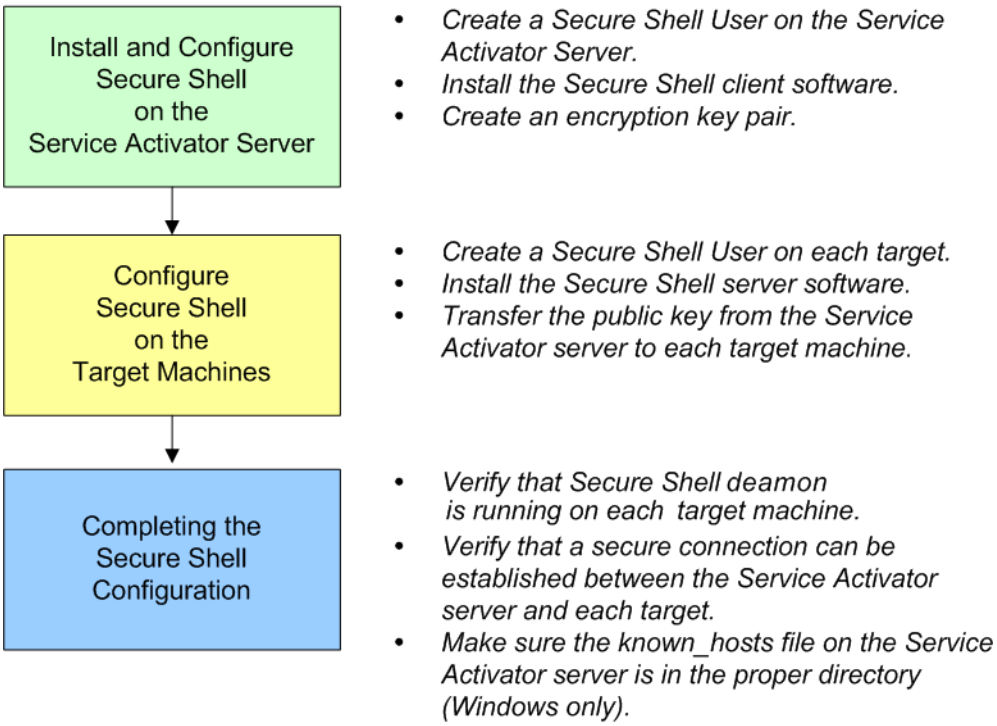
After installing, set the `JAVA_HOME` environment variable to the JDK install location, and add `$JAVA_HOME/bin` to the beginning of the `PATH` environment variable.

Installing and Configuring Secure Shell

In its default configuration, Service Activator uses Secure Shell, based on Open Source, for secure communication from the Service Activator server to target machines. This means that minimally, a Secure Shell *client* must be installed on the Service Activator server, and a Secure Shell *server* must be installed on all target machines. In addition, the Secure Shell encryption keys that are used to verify remote login must be appropriately distributed between the Secure Shell servers (target machines) and the Secure Shell client (Service Activator server).

The Secure Shell installation and configuration process consists of three primary steps, as shown in Figure 1-1. Each of these steps are described in detail in this section, and troubleshooting tips are provided.

Figure 1-1 Secure Shell Installation Process



For more information about Secure Shell, refer to the information available online at `www.openssh.org`.

NOTE Target machine and Service Activator server can be installed and run on one machine. To achieve this, run `ssh-user-config` and `ssh-host-config`.

WARNING **Early versions of Secure Shell, particularly those using Secure Shell Protocol 1, are known to have security vulnerabilities. For this reason, the following instructions assume Protocol 2. Be sure you keep your Secure Shell versions up-to-date and appropriately patched.**

Installing and Configuring Secure Shell on Your Service Activator Server

The first step in the process of establishing a Secure Shell solution is to install and configure Secure Shell on your Service Activator server. To do this, you must first create a dedicated Secure Shell user on your Service Activator server. Then, you must install the Secure Shell client software on your Service Activator server. Finally, you must configure the Secure Shell client to recognize your new user.

Creating A Dedicated Secure Shell User

You must create a dedicated Secure Shell user that will exist both on the Service Activator server and all the target machines. This will be the user that Service Activator uses to log on to remote systems. The user must be consistent across all Service Activator targets and must exist on a *particular* target before activations can be performed on that target.

NOTE

In the instructions that follow, the Secure Shell user will be called **ovactusr**. You may use any user name you like, as long as that name is the same on your Service Activator server and all your target machines. When using the GenericCLI plug-in, however, it is possible to use different user names for different target machines. All users must exist on the Service Activator server and must be configured with SSH keys etc.

Create the Secure Shell user on your Service Activator server by performing these steps:

1. As **root** on the Service Activator server:
   a. Type **`useradd -m ovactusr`**
   b. Type **`passwd ovactusr`**, and follow the prompts to create a new password for this user. Choose any password you would like—it need not be consistent on all target machines.
   c. Grant **`ovactusr`** super user privileges using `sudo` or a similar mechanism.
2. Log off the Service Activator server. Log on again as **ovactusr**.
3. Verify that `$HOME` is set to **ovactusr's** home directory (for example, `/home/ovactusr` or `/users/ovactusr`

You have now created a dedicated Secure Shell user named **ovactusr** on your Service Activator server.Next, you will install the Secure Shell client software.

Installing the Secure Shell Client on Your Service Activator Server

A pre-compiled Secure Shell solution, both server and client, is available on the Linux kit DVD or CDROM. It is highly recommended to use this version to ensure to get all automatically security updates.

You have now installed the Secure Shell client software on your Service Activator server. Next, you will need to create encyption keys for **ovactusr**.

Creating encryption keys for ovactusr

1. Log on to your system as **ovactusr**.
2. Type **`cd $HOME`**
3. Type **`ssh-user-config`**

You have just created an encryption key pair for **ovactusr** on your Service Activator server.Next, you will configure Secure Shell on each of your target machines. After that, you will test the connections between your Service Activator server and your targets.

Configuring Secure Shell on Your Target Machines

Next, you must configure each of your target machines to be Secure Shell servers. Then, you must make the public key for your Secure Shell user on your Service Activator server accessible to each of your targets.

A Secure Shell server is required on each activation target machine, whether it is Windows, HP-UX, Solaris, Red Hat Linux, or another supported platform. Table 1-1 lists three precompiled, production-quality Secure Shell packages that are currently available for download. You can use any of these packages with Service Activator, or you can use another production-quality Secure Shell package of your choice.

Table 1-1

Operating System	Secure Shell Server Download Site
Windows	`http://cygwin.com`
HP-UX	A pre-compiled Secure Shell solution, both server and client, is availabe on the HP-UX install media.
Linux	A pre-compiled Secure Shell solution, both server and client, are included by default.

For information about Secure Shell packages for other platforms, see `openssh.org`.

UNIX Targets

Installation

See the online instructions provided at the Secure Shell package download site.

Configuration

Perform the following steps on each UNIX target machine in your environment:

1. As **root** on the target machine:
   a. Type **`useradd -m ovactusr`**
   b. Type **`passwd ovactusr`**, and follow the prompts to create a new password for this user. You can use any password you like; passwords need not be consistent across target machines.
   c. Grant **ovactusr** super user privileges using `sudo` or a similar mechanism.
   d. Log out.
2. Log on to the target machine as **ovactusr**.
3. As **ovactusr** on the target machine:
   a. Create a new directory called `.ssh` under **ovactusr's** home directory. Set the permissions on this directory to 700.
   b. The public key on the Service Activator server is stored in two files in **ovactusr's** `.ssh` directory: `id_dsa.pub` and `id_rsa.pub`. Transfer the contents of these two files into a file called `authorized_keys2` in **ovactusr's** `.ssh` directory on the target machine.

NOTE If you have previously configured Secure Shell on this target machine, the `authorized_keys2` file will already exist. If it does exist, append the contents of `id_dsa.pub` and `id_rsa.pub` to the existing information in the file. Do not overwrite the existing information.

   c. Set the permissions on your `authorized_keys2` file on the target machine to 600.
   d. For UNIX targets, it is important that **ovactusr** use `ksh` as the default shell. This is because Service Activator exports environment variables using `ksh` syntax. Check the `/etc/passwd` file, and verify that **ovactusr** is using `ksh` as the default shell.
   e. *(Optional, recommended)* Create the file `$HOME/.ssh/environment`. Add the line `PATH=[`*`path directories`*`]`, where the included path directories should minimally be `/bin:/usr/bin:/sbin:/usr/sbin` as well as any other paths required for execution of commands on the target machine (such as the path to the `sudo` executable if it is being used for super-user privileges on the target machine). This is recommended, since Secure Shell by default only inherits the `PATH` environment with which the Secure Shell and `sshd` executables were compiled. This `PATH` environment may not necessarily be inclusive enough for execution of all standard commands. If you want to know what the `PATH` variable was when the Secure Shell and `sshd` executables were compiled, this information can be found in the `sshd_config` file on the target machine.

Windows Targets

Perform the following installation and configuration steps on each Windows target machine in your environment:

NOTE If the target machine already has Cygwin installed and properly configured for **ovactusr**, proceed to the "Configuration" process below. Otherwise, perform the procedure outlined under "Installation" to install Secure Shell on this target machine.

Installation

An OpenSSH-based server and client is available for Windows platforms through Cygwin, an open source collection of tools that allows Unix applications to be compiled and run on a Windows operating system. HP recommends that you download and install the latest distribution of Cygwin from the `http://cygwin.com` web site to be sure that you have the most recent security patches. Ensure that the following categories/packages are selected:

Table 1-2

Category	Package
Admin	cygrunsrv
Base	all
Net	openssh

Configuration

Once **ovactusr** exists and a Secure Shell solution is installed on your Windows target machine, perform the following configuration steps:

1. Log on to your system as **`ovactusr`**. Be sure to choose your local computer in the box labelled `Log on to:`, and not a domain on your network. Your local computer is usually marked `(this computer)` in the drop-down list.
2. In a command line window (`cmd.exe`), type the following commands:

   `cd c:\cygwin`
   `cygwin`
3. On Windows 2008 a `sshd_server` user is needed. The user is created automatically by the installation script. The password must be entered manually. Please enter the password when prompted (i.e. activator).
4. Type the command **`ssh-host-config -y`**
5. Hit **Enter** at the `CYGWIN=` prompt.
6. Start the `CYGWIN sshd` service. To do this, type `cygrunsrv.exe -S sshd` from the command line.
7. The public key on the Service Activator server is stored in two files in **ovactusr's** `.ssh` directory: `id_dsa.pub` and `id_rsa.pub`. Transfer the contents of these two files into a file called `authorized_keys2` in **ovactusr's** `.ssh` directory on the target machine.

NOTE

If you have previously configured Secure Shell on this target machine, the `authorized_keys2` file will already exist. If it does exist, append the contents of `id_dsa.pub` and `id_rsa.pub` to the existing information in the file. Do not overwrite the existing information.

8. For Windows targets, **ovactusr** must use the `bash` shell. Check the file `C:\cygwin\etc\passwd`, and verify that **ovactusr** is using the `bash` shell as the default shell.

Completing the Secure Shell Configuration Process

Log on to your Service Activator server as **ovactusr**.

Perform the following steps for each target machine in your environment:

1. If the target is a UNIX machine, verify that the `sshd` process is running. If the target is a Windows machine, verify that the `CYGWIN sshd` service is running.
2. Type the following command:

   `ssh -l ovactusr -i /home/ovactusr/.ssh/id_rsa <target_machine_name>`

   where *<target_machine_name>* is the fully qualified domain name or IP address of the target machine. Answer **`yes`** when prompted to accept the host key.
3. Type **`exit`** to end this Secure Shell session.
4. Type the following command:

   `ssh -l ovactusr -i /home/ovactusr/.ssh/id_dsa <target_machine_name>`

   where *<target_machine_name>* is the fully qualified domain name or IP address of the target machine. Answer **`yes`** if you are prompted to accept the host key.
5. Type **`exit`** to end this Secure Shell session.

Repeat steps 1-5 for every target machine in your environment. This will populate the `known_hosts` file on your Service Activator server with the public keys of your target machines. If you intend to use the old CommandLineScriptDeployer where SSH is used from the command line, then you need to append the content of the `known_hosts` file that you have just populated in the **ovactusr** .ssh catalog into the `known_hosts` file located in `/.ssh`. This must be done as root.

Installing Database software

Use the documentation provided with Oracle or PPAS to install the database software. During the installation of PPAS it is important to configure PPAS to run in “Oracle” mode. After it is installed, complete the following steps:

1. Create a database user/instance for use by Service Activator (1 GB minimum recommended).
2. If Oracle is used, create a Service Activator user (*`$ACTIVATOR_DB_USER`*) with `dba` permissions in the SID.

NOTE

If you want to run you Oracle database in a way that does not require database administrator ("dba") privileges then please See “Running Service Activator with a Non dba Oracle User” on page 27.

3. If PPAS is used, create a user, a database, and make the user the owner of the database. Also, edit the file `pg_hba.conf` to ensure that the database can be accessed remotely. Finally edit the file `postgresql.conf` and set the parameter `default_with_rowids` to `on`. Also you may want to set the parameter `max_connections` if the default number of connections (100) is not enough. Restart PPAS when configuration is applied.

4. If Oracle is used, start up a listener (if one is not currently running) for the SID, bound to the default port 1521 .

Setting Up the Database Instance and Listener to Start Upon Boot (Oracle only)

If you are running the database on the same machine that runs Service Activator, and you have set Service Activator to start automatically at boot, you will also want the database instance and listener to start upon boot:

1. Create a file named `dbora` using the instructions provided in your Oracle installation guide. You can also find these instructions at `http://docs.oracle.com`.
2. Place the `dbora` file you created into the `/etc/init.d` directory
3. Determine the start order for the Oracle database using the start and stop order for Service Activator as your guideline. The start order is Oracle, Service Activator. The Service Activator start number is 98 on Linux.As shown in Table 1-3, use a lower number as the Oracle start number

Table 1-3

Start Number	Stop Number
less than 98 (for example 97)	100 - Oracle_start_number (for example, 7 (100-93))

4. Create the following links. Replace "***X***" with the start number and "***Y***" with the stop number you determined in step 3:

   `ln /etc/init.d/dbora /etc/rc3.d/SXdbora`

   `ln /etc/init.d/dbora /etc/rc2.d/KYdbora`

5. Reserve port 1521 in `/etc/services` for the Oracle listener by adding the following line to the end of the file:

   `listener 1521/tcp #Oracle listener`

Running Service Activator with a Non dba Oracle User

In order to create an Oracle user for HP Service Activator, you need to do the following:

- Create a new role to be used by the Service Activator database user (this step only has to be executed once)
- Use the newly created role every time you create a new Service Activator database user

The two steps are described in detail below.

NOTE You may want to use a configuration that is different from the configuration described in this chapter. For instance, you may want to create a table space with a name that is different from USERS or you may want to set a quota on the maximum amount of data that Service Activator is allowed to store. Hence, you should see the information in this chapter as an example of how you can create you Service Activator database user.

The privileges that are suggested are adequate for running Service Activator as such. However, Service Activator solutions may require additional privileges. If that is the case, you need to add more privileges to the Service Activator role that is created.

Creating New Role

Run the following SQL as database administrator on the Oracle database server to create a new role that can be used for new Service Activator database users:

```
-- ********************************************
-- Create a role called "hpsa_role"
-- ********************************************
create role hpsa_role;

-- ********************************************
-- Allow user to connect to DB
-- ********************************************
grant create session to hpsa_role;

-- ********************************************
-- Allow the following operations:
-- create/modify/drop tables
-- create comments
-- create indexes
-- create constraints (pk/fk)
-- ********************************************
grant create table to hpsa_role;

-- ********************************************
-- Allow user to create and drop sequences
-- ********************************************
grant create sequence to hpsa_role;

-- ********************************************
-- Allow user to create and drop triggers
-- ********************************************
grant create trigger to hpsa_role

-- ********************************************
-- Allow user to create and drop views
-- ********************************************
grant create view to hpsa_role
commit;
```

Now, a new role called "hpsa_role" with adequate permissions for running HP Service Activator has been created.

Creating a New Service Activator Database User

Run the following SQL as database administrator to create a new Service Activator database user:

```
-- ********************************************
-- Create new user with unlimited quota
-- ********************************************
create user <USERNAME> identified by <PASSWORD> default tablespace USERS quota
unlimited on USERS;

-- ********************************************
-- Assign the "hpsa_role" to the newly created user
-- ********************************************
grant hpsa_role to <USERNAME>;
```

You need to replace <USERNAME> and <PASSWORD> with the actual user name and password, respectively.

NOTE This is just an example. You may need another database configuration in your set up.

Oracle Real Application Clusters (RAC)

Service Activator can also run with Oracle Real Application Clusters (RAC).

The only difference that you need to be aware of is that when ActivatorConfig gets to the "Oracle Database Configuration" screen, you must enter all Oracle RAC hosts - using commas as separators - in the "Host" text field.

All tools that are shipped with Service Activator can be used with an Oracle RAC database. The following tools will always use the database that you have configured with ActivatorConfig:

- DeleteCompletedTransactions
- ViewTransactions
- modifySystemPassword
- servicebuilder ("Service Builder")

Some tools, however, allow the user to specify an alternative database. These tools are:

- designer ("Workflow Designer")
- deploymentmanager ("Deployment Manager")
- InventoryBuilder
- InventoryTreeDeployer
- inventoryTreeDesigner

With the five latter tools you can work with an alternative Oracle RAC database by passing a comma-separated list of database hosts.

Installing a Supported Browser

Make sure that all users who interact with Service Activator use a supported browser only.

Set the following values for your Display properties:

- Screen area: 800x600 (minimum); 1024x768 (recommended).
- Colors: 16 bit (minimum); 24 bit (recommended).
- Fonts: normal size; 96 DPI (recommended).

If you use values other than those indicated here, the browser will not display many of the frames and colors properly in the Operator UI.

2 Installing the Product

This chapter provides the instructions for installing HP Service Activator on the Linux operating system. Before installing Service Activator, be sure that your system meets the hardware and software requirements detailed in “Understanding Software Requirements for the Service Activator Server” on page 16.

Getting Started

Use the instructions in this section to install Service Activator on the Service Activator server.

The Service Activator CD is organized as follows:

`/Binaries/Unix/` Location of the UNIX installation depots for Service Activator, Auto JBoss, and the UNIX install script

`/Binaries/Windows/` Location of the Windows installation file, `ServiceActivator.exe`, which also includes Auto Pass licensing and JBoss

`/Documentation/` Location of the product documentation

`/ReadMe/` Location of end user license agreement as well as 3rd-party licenses

`/Opensource/` Location of 3rd-party sources

Preparing for Installation and Configuration

When you install the product, verify the following:

- `$JAVA_HOME` is set to point to the location of the Java JDK.

Have the following information available:

- The ports that the Workflow Manager, Oracle Database Listener/PPAS, Resource Manager and Web server will use to communicate. Default ports are:
  - 2000 for the Workflow Manager
  - 1521 for the Oracle Database Listener
  - 5444 for PPAS
  - 9223 for the Resource Manager
  - 8080 for the Web server
- The virtual IP address configuration in case a virtual IP address should be set up. This include:
  - Virtual IP address
  - Subnetwork mask
  - Interface - the name of the interface where the IP address will be configured
- The Single Sign On (SSO) configuration parameters. By configuring SSO a user logs in once and gains access to all systems without being prompted to log in again at each of them. It is relevant to configure SSO when cross launching between different independent applications is going to be used. The parameters must match with the configuration for the other applications. SSO on Service Activator is based on LightWeight Single Sign On (LWSSO) and can be used together with other applications which also support LWSSO. The following configuration parameters must be provided:
  - Domain - the domain where Service Activator is running

  - Cipher type
  - Cipher algorithm
  - Key size
  - Init string - encryption string
  - Session timeout
  - Protected domains - the domains which Service Activator should be able make cross lauch to and where other application can be make cross launch to Service Activator from
  - LWSSO log directory

- The disaster and recovery configuration parameters. When configuring a Service Activator cluster node you must indicate if this node belongs to the primary site or one of the standby sites.
  - Site name - The information is used to collect the cluster nodes into different sites. The primary site is the one used in daily operations and the standby sites are sites which can take over the primary sites work in case of disaster.

  When a standby site is configured the database configuration parameter for the primary site must be given in order to register the standby site in the primary site's system database.
  - Username
  - Password
  - Database host
  - Database instance
  - Oracle Listener or PPAS database port (default is 1521 for Oracle and 5444 for PPAS)
- The account and port information you used when creating the database instance and user account during the Database configuration, including:
  - Username
  - Password
  - Database host
  - Database instance
  - Oracle Listener or PPAS database port (default is 1521 for Oracle and 5444 for PPAS)
- The System User configuration:
  - System User name
  - System User password
- The Secure Shell account user name, identity file, and the directory in which the Secure Shell executable resides.
- Your HP Order Number for AutoPass licensing.

Migration from an Old Version

If you wish to upgrade an old Service Activator installation to this version, then please look in the “Migration Guide” for further information.

Preserving Configuration Files

You should not attempt to reinstall Service Activator over an existing installation. If you wish to replace an existing installation for any reason, you should first uninstall Service Activator and then perform a new installation. Before doing so, however, you may wish to backup the following configuration files to a location outside the Service Activator and JBoss installation directories. In particular, if you have customized any of these files, and you would like to reuse the customized information in your new installation, be sure to backup these files.

- *$ACTIVATOR_ETC*/config/mwfm.xml—the configuration file for the Workflow Manager
- *$ACTIVATOR_ETC*/config/resmgr.xml—the configuration file for the Resource Manager
- *$ACTIVATOR_ETC*/config/service_builder.xml—the configuration file for Service Builder
- *$ACTIVATOR_ETC*/config/dm.xml—the configuration file for Deployment Manager
- *$JBOSS_DEPLOY*/hpsa.ear/activator.war/WEB-INF/web.xml—stores the configuration for the UI and servlets
- *$JBOSS_DEPLOY*/hpsa.ear/deployer.war/WEB-INF/web.xml—stores the configuration for the deployer servlet used by Service Builder
- *$JBOSS_HOME*/standalone/configuration/standalone.xml—stores the configuration properties for connecting to the database

NOTE Any Service Activator components installed in the *$JBOSS_DEPLOY* directory are backed up to the *$JBOSS_HOME*/standalone/deployments/deploy.hpsa.bak directory when you reconfigure Service Activator using the ActivatorConfig program. For additional information about ActivatorConfig, see “Configuring Service Activator” on page 35.

Installing

The installation process for Service Activator includes installation of the Service Activator software (including the JBoss application server), configuration of the software, and installation of AutoPass licensing. These procedures are described in the following sections.

Installing Service Activator

1. Verify that all of the pre-installation requirements have been met (see the previous chapter).
2. As root, mount the Service Activator installation compact disc.
3. Change to the `/Binaries/Unix` directory.

NOTE

You must install Service Activator through an XWindows connection because the installation requires GUI interaction. If you are connected to a remote machine, you must ensure that your `DISPLAY` environment variable is set to point at your local machine, and that you have allowed access to the remote machine via the `xhost` command. For example, if the machine you are installing Service Activator on is named `machine2`, your display machine is `machine1`, and you are using Korn Shell, you would run:

```
machine2: export DISPLAY=machine1:0.0

machine1: xhost machine2
```

4. Run the `install` script from the `/Binaries/Unix` directory to install the product bits in the designated locations.
5. After Service Activator have been installed, the installer starts the AutoPass installation software called *AutoPass: License Management* to install the Service Activator license.

   The documentation for AutoPass is reached through the help menu entry.

   If you exit the AutoPass program by clicking the cross, AutoPass provides a 180-day trial password to allow you to immediately begin using Service Activator.

Configuring Service Activator

If you want to configure Service Activator without bringing up a graphical user interface then please also read the section “Configuring Service Activator using configuration file” on page 37.

NOTE

You can reconfigure Service Activator at any time after the initial installation by running the `ActivatorConfig` program located in the *`$ACTIVATOR_BIN`* directory. Be aware, however, that there are several template files that `ActivatorConfig` uses as it configures or reconfigures the product. These include `mwfm_template.xml` and any other files with the word “template” in the file name. If you remove these files, you will not be able to reconfigure your installation properly. Before you rerun `ActivatorConfig`, see “Preserving Configuration Files” on page 34 for instructions about backing up these and other important files.

Before you reconfigure Service Activator, shut down Service Activator (see “Starting and Stopping Service Activator” on page 42).If this is the first time you have installed Service Activator, this component will not be running.

Follow these steps to continue installing Service Activator:

1. Run the `ActivatorConfig` program located in the *`$ACTIVATOR_BIN`* directory.
2. Click `[OK]` to continue past the Welcome screen.
3. `ActivatorConfig` first looks for any existing Service Activator web application components. If it finds these components, it gives you two options:

   - Complete configuration.
   - Partial Configuration.

   If you choose the “Partial configuration” then you can select multiple components to be configured. For each component which is selected the configuration UI will be almost identical to when you do a complete configuration. When this option is selected the existing configuration files are updated and not rewritten from the templated files.
4. When doing a “Complete configuration” `ActivatorConfig` again looks for any existing Service Activator web application components. If it finds these components, it gives you two options

   - Backup and replace the existing Service Activator web application components.
   - Update the existing Service Activator components.

   If you choose the “backup and replace” option, `ActivatorConfig` copies both your web application configuration files and the complete HP Service Activator web useer interface into the directory *`$JBOSS_HOME`*`/server/default/deploy.hpovact.bak`. After you complete the Service Activator configuration process, you can manually merge these files back into your installed Service Activator directory structure.

   If you choose the “update” option, `ActivatorConfig` leaves your the complete HPService Activator web user interface in place and only backs up and replaces your configuration files. You can manually merge your existing configuration files into your installed Service Activator directory structure later if you like.

   See “Preserving Configuration Files” on page 34 for more information about backing up configuration files.

   Click `[Next]` to continue.
5. Specify the database vendor, possible options are Oracle and EnterpriseDB (to use PPAS). Then specify new ports, or use the default ports for the Workflow Manager, the Resource Manager, the Oracle database listener/PPAS database port, and the Web server, and then click `[Next]`.
6. Specify a virtual IP address in case you would like to have Service Activator also to setup a virtual IP address while it is running. The virtual IP address will be setup when Service Activator is started and will be brought down when Service Activator is stopped. In case of cluster node crash the IP address will be taken over by one of the other cluster nodes.
7. Define the user and password for accessing the JBoss Management Console.

8. Specify the parameters needed to Single Sign On in case you would like to configure Service Activator to use SSO. It is only necessary to configure SSO in case you are going to cross lauch to or from Service Activator to other applications which support LightWeight Single Sign On.
9. Specify the site name for which this cluster node belongs. In case you are configuring a cluster node in a standby site the you also need to specify the credentials to the database server running on the primary site.
10. Specify the parameters needed to gain access to the database where Service Activator will store activation elements. The information you provide must be consistent with the database user you created earlier.
11. If you are installing for the first time, make sure the `Create Database` check box is selected so that the Service Activator database tables are created. This check box is selected by default. If you are installing the second node in a cluster environment or if you are reinstalling and would like to use database tables created with a previous installation, be sure to clear the `Create Database` check box. If you are reinstalling and do not wish to use the Service Activator database tables you previously created, you must first manually delete the tables.

    Click `[Next]` to continue.
12. Specify the system user and password. This page will only be shown if the database tables are also created. The system user is used for all internal communication, e.g. for communication between cluster nodes. The system user must exist even when authorization is disabled. If an authorization module is used which bases its authorization on the operating system the user must also be created there.

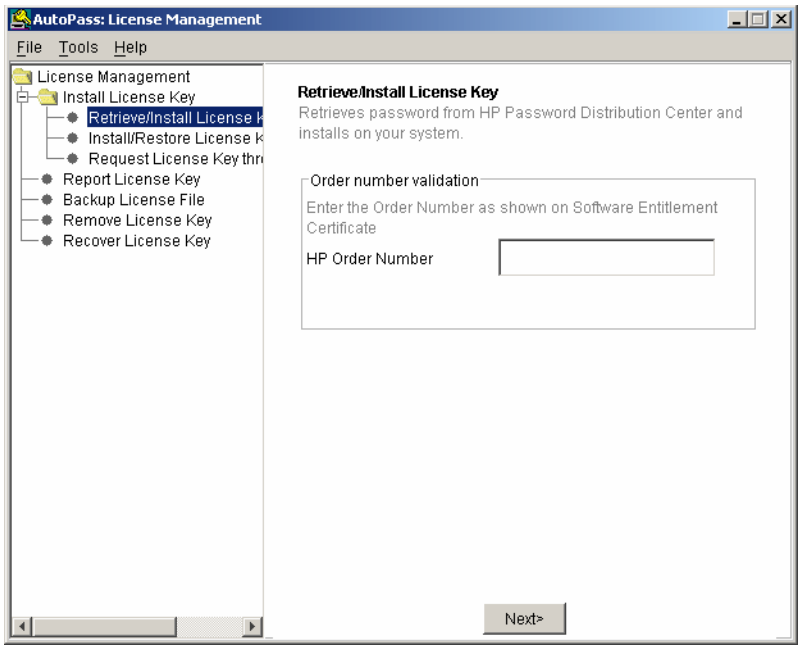
    The system user will always be created with the roles "admin" and "internal"

    Click `[Next]` to continue.
13. In the next dialog box, specify the Secure Shell user name you created for this installation (*$ACTIVATOR_SSH_USER*).
14. Specify or browse to the Secure Shell configuration identity file, which will have the name `identity` for Secure Shell Protocol 1 configurations, or `id_dsa/id_rsa` for Secure Shell Protocol 2 configurations. Be sure *not* to select the `identity.pub`, `id_dsa.pub`, or `id_rsa.pub` file, as this will cause activations to fail. It is recommended always to use Protocol 2 when possible.
15. Specify or browse to the Secure Shell `bin` directory where the Secure Shell program resides, and then click [OK]. Note that if you are browsing to the directory, you must *select* the directory where `Secure Shell` resides in the file selection dialog, as opposed to browsing *into* the directory where it resides.
16. After the configuration is complete, click `[Finish]`.

Configuring Service Activator using configuration file

Alternatively to using ActivatorConfig's graphical user interface, you can configure Service Activator by specifying all configuration parameters in a configuration file. A template file called activatorConfig_template.xml can be found in the directory `$ACTIVATOR_ETC/config`. ActivatorConfig is called in the following way to read the parameters from a configuration file:

```
# $ACTIVATOR_OPT/bin/ActivatorConfig -f $ACTIVATOR_ETC/config/<filename>
```


AutoPass: License Management
File Tools Help
License Management
Install License Key
Retrieve/Install License K
Install/Restore License K
Request License Key thr
Report License Key
Backup License File
Remove License Key
Recover License Key
Retrieve/Install License Key
Retrieves password from HP Password Distribution Center and installs on your system.
Order number validation
Enter the Order Number as shown on Software Entitlement Certificate
HP Order Number
Next>

Running Service Activator as a Non Root User

Service Activator can be setup to run as a non-root user on Linux operating system. However you need to always install Service Activator as root, but you do not need to run ActivatorConfig.

NOTE When running Service Activator as a non-root user, there will be some limitations imposed by the operating system that solution developers need to take into account when developing a solution for Service Activator. For instance, it will not be possible to bind to a port number lower than 1024 or write files to locations outside Service Activator's directories (except for the /tmp and the user's home directories). If binding to a port number lower than 1024 is required by your solution, it is recommended to run Service Activator as the user "root".

Before configuring Service Activator to run as a non-root user, you must first create an appropriate user and (optionally) group on the UNIX system. Consult the documentation that came with your UNIX system to find out how to create new users and groups.

After you have created a new user you need to stop Service Activator if the process is running.

After Service Activator has shut down you need to run the script `AssignNonRoot` located in the `$ACTIVATOR_OPT/bin` directory. This script must be run as the user "root".

The script will prompt for the user name and group name of the user that will be used for running Service Activator.

After the AssignNonRoot script has completed (which can take some minutes), Service Activator can be started in the usual way.

NOTE Even if you run the Service Activator startup script as "root", the Service Activator process will actually run as the newly created user.

Every time you install a new hotfix or service pack for HP Service Activator you need to run the AssignNonRoot script again.

Files Deployed using the Inventory Builder and the Deployment Manager

When deploying files using the Deployment Manager or Inventory Builder on UNIX, the permissions of the deployed files will be determined by the umask (abbreviated from "user mask") that is set for the user running the Deployment Manager or Inventory Builder; hence, the permissions of the source files will not be preserved when they are copied to another location.

So, for instance, if a user has the umask set to 022, all files that are created (by the Deployment Manager or Inventory Builder) will have the permissions 644 and all directories will have the permissions 755.

NOTE Most UNIX systems do not allow new files to be created with execute permission turned on, regardless of the umask.

When running Service Activator as non-root, it is important that the user's umask is set to 0xx, where x can be any value from 0 to 7. The recommended value of umask is 022.

Using Secure Socket Layer (SSL) with Service Activator

You can configure the Service Activator Operator UI to use Secure Socket Layer Protocol (SSL) for HTTPS. You can also configure the Workflow Manager to use SSL when sending messages to and receiving messages from a Customer Relationship Management (CRM) system. For additional information, see Appendix E, “Configuring Service Activator to Use Secure Socket Layer (SSL) Protocol,” on page 69.

Starting and Stopping Service Activator

You can start and stop Service Activator the following way:

Use the `activator` script located in the `/etc/init.d` directory. It has the following usage syntax:

`activator` { start | start_interactive | stop | check }

Securing Administration Consoles

JBoss application server comes with a web administration console. The console provides a view of the JBoss application server. It lists all registered deployment that are active in the application server, datasources, and port configuration.

It can be accessed from your web browser `http://localhost:9990/console`. When installing Service Activator the console is configured in such a way that it can only be accessed from localhost. The user name and password is configured as part of the Service Activator configuration.

More information about administration consoles can be found at http://www.jboss.org

Change Memory Allocation

Service Activator has predefined memory allocation. By default, the minimum memory allocation is 256Mb, maximum – 1024Mb. In this section, it is described how to increase or reduce memory size for each of the components. It is strongly recommended to leave minimum memory allocation set to 256Mb. Maximum allocation size limit must be equal or greater than the minimum size.

It is possible to change memory allocation for Service Activator. This done by changing the variables `JVM_MIN_MEMORY`, and `JVM_MAX_MEMORY` in the JBoss standalone configuration script. These variables specify the minimum and the maximum memory allocation. The JBoss standalone configuration script can be found in the directory `$JBOSS_HOME/bin` and is named `standalone.conf`. After the change Service Activator must be re-stared

In same way it is also possible to change the permanent generation memroy size. The variable to handle this is called JVM_MAX_PERM_SIZE. Service Activator is per default configured with the size 384Mb.

Starting the Operator UI

To start the Operator UI, start Service Activator, then open Internet Explorer, and go to the following URL:

`http://<machine_name>:8080/activator/login.html`

NOTE You need to use the full qualified domain name of your Service Activator server in case you have configured Service Activator to run with Single Sign On.

In the login screen, enter the user/password of the user authorized to log into Service Activator. Because this is an initial installation, the OS Authentication module has not yet been activated. Consequently, you can login with any user/password. Refer to HP Service Activator—Workflows and the Workflow Manager for additional information about authentication.

Go to the logs screen to view the various log files and verify that everything is working as expected.

For more information about the Operator UI, see the *"User's and Administrator's Guide"*.

Removing the Installation

The Service Activator product should be removed only by using the remove script. This allows the software distribution database, a database of the products that have been installed on the machine, to be updated. After properly removing Service Activator, you can safely remove any directories or files that are left.

Use the following instructions to remove Service Activator:

1. As root, run:

   `/opt/OV/ServiceActivator/bin/remove.serviceactivator`

2. This leaves several database tables that you must remove manually. Log in to the database machine and type the following statements:

NOTE

When Service Activator is removed, generated and customized files are not deleted by the `remove.serviceactivator` script. This means that the *`$ACTIVATOR_OPT`*, *`$ACTIVATOR_VAR`*, *`$ACTIVATOR_ETC`*, and *`$JBOSS_HOME`* directories will not be empty and, therefore, will not be automatically removed. If you do not want to save any of the customized or generated files, you can manually delete these directories *after the standard removal process completes*.

Internet Protocol versions IPv4 and IPv6

HP Service Activator supports both IPv4 and IPv6. So when configuring Service Activator a decision must be taken if the internal communication between the cluster nodes as well as the communication with the database should be using IPv4 or IPv6. For this kind of communication the same IP protocol stack must be used. However the virtual IP addresses can be setup with another protocl stack so e.g. Service Activator internally is using IPv4 but the users are accessing via IPv6.

The virtual IP addresses may or may not belong to the same IP subnet ast the cluster node's primary IP addresses. However, if the virtual IP addresses ant the primary IP addresses belong to different subnetworks it is important that each cluster nodes is configured with additional fixed IP address in the same subnetwork as the virtual IP addressses. Otherwise, the virtual IP functionality will not work as designed.

It is also possible to use HP Service Activator to communicate with target equipments using both IPv4 and IPv6 addresses.

Using Service Activator in a Cluster Environment

It is possible to run Service Activator in a cluster environment. When doing this there are a number of things which need to be considered:

- Where to have the Oracle Database Server
- Which kind of authentication and authorization should be used
- Which clients must connect to Service Activator
- Location of servers
- Configuration files
- What kind of load balancing scheme should be used

Oracle Database Server

The Oracle database server should run on its own platform in the case where Service Activator is running in a cluster environment. To get a full reliable system the Oracle Server must also run a high availability environment. Service Activator expects that it can get in contact with the Oracle database using same IP address at all times

Authentication and Authorization

Five different authentication modules are provided with Service Activator. Four of them are based on the underlying operating system whereas the fifth is a database authentication module. The same module must be configured on all cluster nodes and if using one of the operating system dependent modules the same users with the same passwords must be created on all cluster nodes. It is much simpler to use the database authentication module as the users only need to be setup one time. Hence using the database authentication module is highly recommended.

Also the roles, operating groups, needs to be the same on all cluster nodes.

Access from Clients to Service Activator

When a client accesses Service Activator it will send its request to one of the cluster nodes and then this will handle the load distribution depending on which distribution module is configured. However if the cluster nodes that the client is contacting is unavailable the client needs to contact one of the other cluster nodes in the system. This can be done in a number of ways:

- "The client holds a complete list of all Service Activator cluster nodes and connects to one of the other cluster node in case the connection is lost
- "A high availability package is installed along with Service Activator on all cluster nodes to provide a virtual IP address

Access to Target Systems

Each cluster node in the Service Activator cluster environment must have access to the same target systems. This is needed since a workflow would automatically failover to another cluster node in case one cluster node fails.

Location of Servers

All servers taking part in a Service Activator cluster must run identical operating systems. In addition, they must have a fixed IP address and must be connected to the same sub-network. The database server must also be connected to the same sub-network. The fixed IP address of the Service Activator server must be the one that matches the server's hostname.

Finally, all cluster servers must be added to the /etc/hosts file on all servers (`%SystemRoot%\system32\drivers\etc\hosts` on Windows). In this way, you can ensure that the Service Activator cluster will not be impacted by a failing DNS server.

Configuration Files

Normally you will have exactly the same configuration on all your cluster nodes. However there might be differences due to unequal processing power of the hardware on which Service Activator is installed. E.g. one of the cluster nodes is much more powerful than the other ones it might make sense to configure this node to handle a higher number of request received from the client side. This can be done by using e.g. the LoadFactorDistributionModule. See "LoadFactorDistModule" on page 391 of *HP Service Activator-Workflows and the Workflow Manager*. It might also be an idea to increase the number of connections to the database server and the number of worker threads configured in the Workflow Manager. However, in normal situations you will have the same configuration on all cluster nodes. The Deployment Manager can be used to identify the difference between the configurations of the nodes in a cluster.

When running in a cluster environment a distribution module must be configured. This needed by the cluster nodes to detect that they running in cluster environment. Apart from that a table in the database, the cluster node list, will also contain one row per cluster node added to the system. By combining this information it is easy for a single cluster node to figure out if it is running in a clustered environment and which other cluster nodes belong to the cluster.

Load balancing scheme

Service Activator comes with tree kind of distribution modules RoundRobinDistModule, LoadFactorDistModule,and QueueDistModule. One of these must be configured when running in a distributed environment. Which one to chose depends very much on the configuration of the system. In most cases, the RoundRobinDistModule would be the easiest and most natural choice. However, if either one of the cluster nodes is much more powerful than the other ones, then it would make sense to use the LoadFactorDistModule. Finally, it would make most sense to use the QueueDistModule in cases where there is a great variety in how much work each workflow performs.

Adding Cluster Node

When installing the first cluster node the installation is done in exactly the same way as in a standalone system which has be described in the previous chapters. The second node is then added by again installing Service Activator and running ActivatorConfig the same way as in a standalone environment, but without having the "Create Database" check mark set. Remember to use the same database user, password, host, instance, port. Then the database tables will not be created but the "cluster node list" in the

database will be updated with the added cluster node. So even in the case where the database tables should not be created ActivatorConfig still requires that it is possible to setup a connection to the database server.

Disaster Recovery Considerations

Service Activator can also run in a disaster recover setup. Such a setup will contain a number of sites where the primary site is the site which is running under normal operation. The other sites are called standby sites and multiple standby sites can exist. The standby sites should not be running during normal operation.

Each site can contain a number of cluster nodes where all the cluster node in one site must be connected to the same lan segment and have access to an Oracle database server at the same lan segment. A standby site will typically be placed in another building or region and will take over the operation in case of problems at the primary site. When a take over is done the new site becomes the primary site.

The database server at the primary site is the only database which is accessed during normal operation i.e. syncronization between the database on the primary site and the backup sites are not handled by Service Activator. This syncronization can be done by means of e.g. Oracle Data Guard.

The take over process for a standby site to be the new primary site is done in the following way:

- Ensure Service Activator is not running at the primary site
- Ensure the standby oracle database is up and running and contains the right data
- Start Service Activator and log in on the user interface as the system user
- Go to the Node Information page
- Right click in the site which you would like to become the primary site and select the option “Make Primary Site”

After this operation the new primary site will take over all existing jobs which were running on the old primary site and the old primary site will now be a standby site.

A Scripts

Appendix A contains the locations and descriptions of the scripts necessary to maintain your Service Activator installation.

Understanding Available Scripts

This table lists the locations and descriptions of the scripts that are available in Service Activator. Unless otherwise indicated, these files are located in the *$ACTIVATOR_BIN* directory.

Table A-1 Service Activator Scripts

Script	Description
`/etc/init.d/activator`	Starts/stops the Service Activator processes (Solaris, linux).
`/sbin/init.d/activator`	Starts/stops the Service Activator processes (HP-UX).
`ActivatorConfig[.bat]`	Configures Service Activator for a specific environment.
`AssignNonRoot`	Configures Service Activator to run as non-root, only on UNIX
AuditFilter	Imports/exports audit filters from/to file.
`CatchSocketSenderMessages[.bat]`	Listens for messages on a given port and prints those messages to `stdout`. This script is typically used for testing and demonstration of the `SocketSenderModule` of the Workflow Manager. By default, it listens on port 4099, but takes a single parameter to specify the port.
`checkLicence.[bat]`	Checks the status of the HP OpenView AutoPass license and prints out debug information.
`CleanLogs[.bat]`	Deletes all but the active logs
`crypt[.bat]`	Encrypts or decrupts a password for local use, to avoid storing unenrypted passwords in the workflow manager configuration file.
`DataSourceConfiguration[.bat]`	Assist in management of data source configuration
`dc[.bat]`	Starts Data Collector, command line tool for gathering information about the Service Activator components (Workflow Manager, Resource Manager, JBoss).

Table A-1 Service Activator Scripts (Continued)

Script	Description
DeleteCompleteTransactions[.bat]	Cleans up saved compledted activtion transactions.
deploymentmanger[.bat]	Runs the Workflow Designer tool
designer[.bat]	Runs the Workflow Designer tool
generateEncryptedPassword[.bat]	Utility to generate an encrypted password. This can be used when an additional data source file has to be created.
generateMD5.[bat]	Calculates MD5 checksum for a a file.
InventoryBuilder[.bat]	Runs the InventoryBuilder tool
InventoryTreeDeployer[.bat]	Runs the Tree Deployer tool
inventoryTreeDesigner[.bat]	Runs the Inventory Tree Designer tool
generateMD5[.bat]	Generates a md5 checksum on a given file
modifySystemPassword[.bat]	Utility to update the system user password. The password must also be changed in the auth module.
mwfmtool[.bat]	A command line tool for performing workflow engine tasks such as starting workflows and viewing posted messages. If this script is executed without any parameters, it will display a list of all the tasks that can be performed.
removeClusterNode[.bat]	Removes the specified cluster node from the clusternodelist in the database.
remove.serviceactivator	Uninstalls Service Activator on UNIX.
servicebuilder[.bat]	Invokes the Service Builder executable, either the command line (if arguments are passed) or the GUI (if no arguments are passed).
TestAtomicTask[.bat]	Starts an atomic task for testing purposes.

Table A-1 Service Activator Scripts (Continued)

Script	Description
UMMData.[bat]	Imports/exports users, roles with Inventory UI privileges from/to file.
updateLicence.[bat]	The script lets you update your trial or existing licence for HP Service Activator.
ViewTransactionState[.bat]	Displays the different states of a completed transaction.
Web Service Designer[.bat]	Runs the Web Service Designer tool

B Configuration Files

Appendix B contains the names and descriptions of the configuration files, some of which can be modified as needed for your installation.

Understanding Available Configuration Files

The following table identifies and describes the configuration files that are provided with Service Activator. Unless otherwise indicated, these files are located in *$ACTIVATOR_ETC*/config.

File	Description
ActivationDialog.dtd	The Document Type Definition (DTD) for for the GenericCLI activation dialog
bean.dtd	The Document Type Definition (DTD) for for the inventory resource definition
CLIv4.dtd	The Document Type Definition (DTD) for for the GenericCLI command dialog
CompoundTask.dtd	The Document Type Definition (DTD) for compound tasks created by Service Builder.
designer.xml	Configuration file for the Workflow Designer.
itd.xml	Configuration file for the Inventory Tree Designer.
dm.xml (dm.dtd)	Configuration file and associated DTD for the Deployment Manager.
deploy.dtd	DTD file for the deployment descripter.
deploy.dtd	DTD file for the deployment descripter.
deploy.dtd	DTD file for the deployment descripter.
install.dtd	DTD file for descriping a solution deployed with the Deployment Manger.
inventoryTree.xml (inventoryTree.dtd)	Configuration file and associated DTD for configuring the structure of the inventory presentation in the Operator UI.
menu.xml (menu.dtd)	Configuration file and associated DTD for configuration of the left navigation menu.
mwfm.xml (mwfm.dtd)	Configuration file and associated DTD for the Workflow Manager.
par.dtd	The DTD for the MANIFEST/par.xml file found in the Plug-in Archive (PAR).
resmgr.xml (resmgr.dtd)	Configuration file and associated DTD for the Resource Manager.
ummData.xsd	the DTD for definiton of User Management Data which can be loaded to the system database.

File	Description
role_mappings.xml (role_mappings.dtd)	Configuration file and associated DTD for configuring mappings from roles that the Workflow Manager is aware of and roles that the authentication module is aware of. The role-mappings.xml is optional.
service_builder.xml	Configuration file for Service Builder.
$ACTIVATOR_ETC/workflows/workflow.dtd	DTD for workflow definition.
$JBOSS_DEPLOY/hpovact.sar/META-INF/jboss-service.xml	Stores the classpath for the Service Activator J2EE components deployed in JBoss. If you do not store your inventory class files in the default location of *$ACTIVATOR*/3rd-party/inventory/classes, you will need to add or modify classpath entries in this file.
$JBOSS_DEPLOY/hpovact.sar/activator.war/WEB-INF/web.xml	Stores the configuration for the UI and servlets.
$JBOSS_DEPLOY/hpovact.sar/deployer.war/WEB-INF/classes/lwssofmconf.xml	Stores the configuration for the LightWeight Single Sign On configuration.
$JBOSS_DEPLOY/hpovact.sar/deployer.war/WEB-INF/web.xml	Stores the configuration for the deployer servlet used by Service Builder. See "Configuring Authentication or Authorization" on page 66 in *HP OpenView Service Activator—Developing Plug-Ins and Compound Tasks* for a description of the configurable parameters in this file.
$JBOSS_DEPLOY/hpovact-inventory-ds.xml	JBoss data source file containing the database name, user, and password used for the Operator UI and inventory.
$JBOSS_DEPLOY/hpovact.sar/hpovact-EJBs.jar/META-INF/jboss.xml	The MaximumSize setting configures the maximum number of concurrent invocations of the EJB that performs task activations. For additional information, see Chapter 4, "Configuring Activation Parameters," on page 87 in *HP OpenView Service Activator—Developing Plug-Ins and Compound Tasks*.
$JBOSS_DEPLOY/hpovact.sar/hpovact-EJBs.jar/META-INF/ejb-jar.xml	Stores the activation time-out configuration parameter. See Chapter 4, "Configuring Activation Parameters," on page 87 in *HP OpenView Service Activator—Developing Plug-Ins and Compound Tasks* for additional information.

File	Description
$JBOSS_DEPLOY/ `hpovact-ra-ds.xml`	The `max-pool-size` setting configures the maximum number of instances of the resource adapter that can be used during task activations. For additional information, see Chapter 4, "Configuring Activation Parameters," on page 87 in *HP OpenView Service Activator—Developing Plug-Ins and Compound Tasks*.
$JBOSS_DEPLOY/ `hpovact-oracle-ds.xml`	Stores the configuration parameters for accessing the Oracle database that stores task information. See the comments in this file for a description of some of the configurable parameters.

C Log Files

This appendix contains the names and descriptions of the log files used by Service Activator and JBoss.

Understanding Available Log Files

Service Activator uses several log files to track events that occur within the product. Many of the logs use an XML-based grammar (those with an `.xml` suffix, listed in the following table). These log files roll over to a new log file once they reach a specific size. You can configure the log size by setting the `log_max_entries` parameter in `mwfm.xml`, and `resmgr.xml`.

The log files are not localized.

All log files are visible in the Operator UI, where you can view the contents of each log file and manage the log files by removing them periodically. When a log reaches a specific size, Service Activator renames the log by appending *`_oldlogfile#`*`.xml` to the log name. Service Activator increments the number of the old log file each time it rolls over a log for that component.

For example, when the `mwfm_active.xml` log file reaches a specific size, it rolls over to `mwfm_0.xml` log file. Similary, when the `_active` file reaches a specific size, it rolls over to `mwfm_1.xml` file, and so on.

The actual log files are located in the `$ACTIVATOR_VAR/log/<HOSTNAME>` directory, where `HOSTNAME` is the computer name on which the Logs are running.

Log files ending in `_active.log.xml` are the active logs for a specific Service Activator component. These are the only log files that are refreshed when the `Auto-refresh` feature is turned on and the user is currently viewing this file. Removing these logs when the component is active can cause Service Activator to fail.

Table C-1 lists the log files currently used by Service Activator.

Example C-1 on page 62 shows the XML DTD for the Service Activator log files, and Example C-2 on page 62 shows a typical `.xml` log entry.

Table C-2 lists a number of JBoss log files that may be useful to you when working with Service Activator. These log files can be found in *`$JBOSS_HOME`*`/standalone/log`.

Table C-1 Service Activator Log Files

File	Description
`connector_active.log.xml`	Provides log entries for transactional coordination of an activation. Also see the `resmgr_active.log.xml`.
`depeng_active.log.xml`	Provides log entries for all deployments of PARs (Plug-in Archives) and compound tasks.
`mwfm_active.log.xml`	Contains the log entries posted by the Workflow Manager during normal operation.
`resmgr_active.log.xml`	Contains the log entries posted by the Resource Manager during normal operation.
`designer.stdout`	Contains `stdout` diagnostics from running the Workflow Designer. Useful when trying to identify problems running the Workflow Designer.
`designer.stderr`	Contains `stderr` diagnostics from running the Workflow Designer. Useful when trying to identify problems running the Workflow Designer.

Example C-1 DTD of XML-based Log Files for Service Activator

```
<!-- The log entry -->
 <!ELEMENT LogEntry (Time, Module, Part, Component, Topic?,
   Thread, ID?, Message)>
 <!-- level of the entry. It can be one of INFORMATIVE,
   WARNING, ERROR, DEBUG, DEBUG2 -->
 <!ATTLIST LogEntry level  CDATA #REQUIRED>
  <!-- The host where the component was running -->
 <!ATTLIST LogEntry machine CDATA #IMPLIED>
  <!-- Date and time this entry was written -->
 <!ELEMENT Time    (#PCDATA)>
   <!-- Specifies which major module of Service Activator
     generate the log entry. The possible values for this
     field are: MWFM, Connector, RM, Inventory, GUI,
     DeploymentEngine, LOGGER_THREAD. -->
 <!ELEMENT Module   (#PCDATA)>
  <!-- Which part of Service Activator generated this entry.
     Valid values are FRAMEWORK (i.e. Micro-Workflow Manager
     or Resource Manager), or COMPONENT (i.e. a workflow
     node, a plug-in, a dynamic module). -->
 <!ELEMENT Part    (#PCDATA)>
  <!-- If the field "Part" is FRAMEWORK, Component is the
     specific module that generated the entry. If field
     "Part" is COMPONENT, this field will indicate the name
     of the component itself (i.e. the name of the workflow
     node or plug-in). -->
 <!ELEMENT Component (#PCDATA)>
  <!-- Topic of the message. Valid values are STARTUP,
     RECOVERY, STATISTICS, and COMMON_OPERATION. -->
 <!ELEMENT Topic   (#PCDATA)>
  <!-- The name of the thread that ran this component
     -->
 <!ELEMENT Thread   (#PCDATA)>
  <!-- Unique identification number for the entry (i.e.
     workflow id, XID). -->
 <!ELEMENT ID     (#PCDATA)>
  <!-- The message itself. -->
 <!ELEMENT Message  (#PCDATA)>
```

Example C-2 Typical Service Activator Log Entry

```
<LogEntry level="INFORMATIVE"
     machine="activator/15.2.114.138">
 <Time>Wed Jul 11 17:08:03 MDT 2001</Time>
 <Module>mwfm</Module>
 <Part>FRAMEWORK</Part>
 <Component>LOGGER_THREAD</Component>
 <Topic>STARTUP</Topic>
 <Thread>main</Thread>
 <Message>logger thread properly started (level
     2).</Message>
</LogEntry>
```

Table C-2 JBoss Log Files

File	Description
`boot.log`	Contains log output from the initial stages of JBoss startup. An error in this log file indicates a problem with your JBoss installation or configuration.
`server.log.*`	Contains log output from the subsequent stages of JBoss startup specifically related to the deployment of applications in the *`$JBOSS_DEPLOY`* directory, including the Service Activator JEE components. An error in this file can indicate either a JBoss-specific problem or a configuration problem with the Service Activator components deployed in *`$JBOSS_DEPLOY`*. If there is an error in this file, verify that any changes made to the Service Activator configuration files.

D Security Considerations

This appendix contains a summary of security information pertinent to Service Activator installations.

Verifying Product Security

Attention to security issues is especially important when you deploy Service Activator in a production environment. The following checklist will help you to ensure that you have configured Service Activator properly to address security considerations.

- Make sure that the Service Activator server is a secure system. To accomplish this, we recommend you obtain a security hardening/lockdown tool that can identify potential vulnerabilities. Listed below, are a few that you should consider:
  - HP-UX 11i v3: HP-UX Bastille (part number B6849AA) for hardening and lockdown. Security Patch Check (B6834AA) to alert you to security-related patches that you should install.
  - Linux: Bastille linux for system hardening.
  - Windows 2008: Microsoft Baseline Security Analyzer for system hardening.
- Set up a secure link between the Service Activator server and the target machines. This is especially important because potentially sensitive information (such as user name and password of an account created on the target machine) is transmitted to the target machine. Only authorized users can perform service activations, so it is important to keep user names and passwords secure.

  To accomplish this, Service Activator employs Secure Shell technology. Instructions for installing and configuring Secure Shell are included in the *HP Service Activator—Installation Guide* located on the Service Activator installation compact disc.
- Set up a secure link between web browsers and the Operator UI. When an operator logs in to the Operator UI, he or she passes user name and password information through the web browser. This communication path must be secure so that the information is not intercepted. See the *HP Service Activator—Installation Guide* for information about securing communication between Apache Tomcat and your web browser.
- Configure the Workflow Manager to perform authentication and authorization. The default configuration of the Workflow Manager disables authorization and authentication checking of users who log in to the Operator UI. Service Activator provides an operating system authentication module that you can use to enable authorization and authentication.

  In addition, Service Activator supports the concept of roles to control which operations a user can perform. Enable the operating system authentication module, and set up roles to control access to various capabilities. See *HP Service Activator—Workflows and the Workflow Manager* for more information.
- Provide secure communication between the OSS environment and the Workflow Manager. Service activation requests passed from the OSS environment to the Workflow Manager may need to be secure because customer-sensitive data (such as user names and passwords) can be passed as part of the activation request. In addition, the communications may need to be secure to ensure that unauthorized agents do not make activation requests. You can address these security requirements by deploying SSL as the communications transport between the OSS environment

and the Workflow Manager. We provide details on how to modify the socket listener module in the Workflow Manager to exploit SSL in *HP Service Activator—Workflows and the Workflow Manager*.

- Configure Service Activator to require authorization and authentication when deploying PARs and compound tasks from Service Builder. The default configuration disables authorization and authentication for deploying PARs and compound tasks. One should note that communications between Service Builder and the Service Activator server cannot be encrypted. If you are running Service Builder in an insecure environment, we strongly recommend you only deploy from an instance of Service Builder residing on the Service Activator server, since authentication information is passed during the deploy step. See "Configuring Authentication or Authorization" in HP Service Activator—Developing Plug-Ins and Compound Tasks for more information about enabling this security.
- The default installation of JBoss has all functionality enabled. If these are potential security issues in the customer environment, we recommend you manually configure JBoss to allow for startup of only selected functionality.
- If you are going to operate Service Builder on HP-UX or Linux via a remote X11 session, you might want to consider using X11 connection forwarding via Secure Shell. This is especially important if you are going to operate Service Activator in an insecure environment. Using X11 forwarding will ensure that authentication information you provide cannot be intercepted. Please consult the following site for instructions for setting up X11 connection forwarding:

  `http://www.itso.iu.edu/howto/ssh2.epl#x11`

- Service Activator uses an Oracle database as an inventory and service repository. Please make sure that you change the Oracle password from its default value to avoid its being compromised.
- Service Activator emits a number of log files containing information about the progress of specific service activations. These log entries can contain information that is specific to an individual customer. Thus, it is extremely important that these files remain secure. The best way to ensure this is to perform system hardening as mentioned earlier.
- Service Activator uses Secure Shell to communicate with target machines. Typically, a firewall will exist between the Service Activator server and the target machines. You will need to allow Secure Shell communications through the firewall. This is the only firewall hole that Service Activator requires.
- When deploying Service Activator in the customer's environment, you need to take into account the fact that the target machines on which Service Activator is operating may be in a different trust domain than the Service Activator server. When Service Activator invokes scripts on the target host, the possibility exists that the target could be hostile and attempt an attack on the Service Activator server based on the information that is returned via the execution of the script. There are two potential areas of concern that you should take into account when developing a solution for the customer:

  — A false return code could be returned by the script in an attempt to affect the branching of the workflow. Normally, this will only affect the activation activity on the target machine in question. If, however, you constructed a workflow to orchestrate activation activities on target machines in two different trust

domains, a false return code generated by a script running on a target in one domain could impact the activities occurring on the other target system if the workflow logic does different things depending on the return code. Depending on the logic in the workflow, you could damage a server based on performing an inappropriate activation activity. Thus, do not construct workflows that will coordinate activation activities that span trust domains.

— Similarly, you cannot necessarily trust the contents of `stdout` and `stderr` information returned by a script. A variety of attacks could be launched if malicious `stdout/stderr` information is provided to other applications involved in the activation request. Exercise caution when using `stdout` or `stderr` information. Detainting (for example, removing escape sequences like a semicolon that could cause the information to be interpreted incorrectly) may be necessary.

- Service Activator does not redeploy scripts to target machines every time they get executed. However, Service Activator restricts access to the scripts so that they cannot be modified. DO NOT modify the access permissions on scripts that have been previously deployed to the target hosts.
- Be careful handling potentially sensitive information to make sure it is not passed as an argument to a script that runs on the target machine. This information can be easily retrieved by getting a list of running processes (for example, “ps -ef” on UNIX). To avoid this problem, take the following steps:

  — Do not pass sensitive information to the script as an argument on the command line. Instead, place sensitive information in a file on the Service Activator server, copy it over to the target machine, read the file contents from the script, and then remove the file.

  — Make sure you do not consequently pass this sensitive information as an argument to other applications.

WARNING **You are strongly urged to avoid using any application that requires you to pass sensitive information to it via command line arguments.**

E Configuring Service Activator to Use Secure Socket Layer (SSL) Protocol

This appendix contains instructions for configuring Service Activator to use Secure Socket Layer (SSL) protocol for HTTPS or for sending and receiving secure messages between the Workflow Manager and a Customer Relationship Management (CRM) system.

Using SSL with Service Activator: An Overview

You can use SSL with two Service Activator components. The first is the Operator UI, which you can configure to use HTTPS. The second is the Workflow Manager, which you can configure to use SSL to send (and receive) secure messages to (and from) a CRM. The configuration processes for both of these components are similar.

Preparing to Use SSL

Implementing a security solution such as SSL is, by nature, a complex process that involves numerous design decisions and trade-offs. This appendix does not attempt to provide a comprehensive discussion of SSL or to offer advice about how best to implement an SSL solution with Service Activator in your environment. It, instead, offers one approach that you can use to configure Service Activator to use SSL.

Before proceeding, you should be knowledgeable about SSL—in particular, using SSL with Java—in order to determine the appropriate SSL solution to use with Service Activator for your environment. The following references can assist you in understanding and implementing an SSL solution:

- The Oracle JSSE web site at `http://java.sun.com/products/jsse`
- The Oracle `keytool` reference at `http://docs.oracle.com/javase/t/doces/technotes/tools/solaris/keytool.html`
- The OpenSSL web site at `http://www.openssl.org`

Getting Organized

Before using SSL with Service Activator, you will need to design a mechanism for using and storing keys and certificates. To do this, you will need to answer the following questions:

- What will you name your keystore?
- Where will your keystore be located?
- What will your keystore password be?
- How and where will you store trusted certificates?
- Will you use client-side authentication?
- Which Service Activator configuration files will you need to update?

Configuring Service Activator to Use SSL

To configure either the Operator UI or the Workflow Manager to use SSL, you will need to complete the following steps:

1. Configure Java Secure Socket Extension (JSSE).
2. Create a certificate keystore.
3. Obtain and import a signed certificate into the keystore.

4. Modify the appropriate configuration files to reflect the keystore name and password.

Component Using SSL	Configuration Files That Require Modification To Use SSL
Operator UI	*$JBOSS_HOME*/standalone/configuration/standalone.xml
Workflow Manager	*$ACTIVATOR_ETC*/config/mwfm.xml

5. Restart Service Activator to ensure that all changes are effective.

Each of these steps will be described in detail for both the Operator UI and the Workflow Manager. For additional information about using SSL with the JBoss/Tomcat bundle, please see *JBoss Administration and Development, Second Edition*. This document is available for purchase at the www.jboss.org web site.

Understanding the Required Software

JSSE is a reference implementation of SSL for Java. It implements the SSL and Transport Layer Security (TLS) protocols. The JAR files for JSSE are supplied by the Java run-time environment (JRE). This package also includes data encryption and server authentication functionality.

Configuring JSSE

In the file named *$JAVA_HOME*/jre/lib/security/java.security, add the following entry if it does not already exist:

```
security.provider.#=com.sun.net.ssl.internal.ssl.Provider
```

Replace the "#" with the appropriate value based on the number of configured providers. It is essential that this value be not only unique, but also sequential starting with the value "1." If you do not comply with this requirement, you will not be able to configure SSL correctly.

Preparing to Load the Certificate Keystore

Tomcat currently only utilizes the Java standard Java Keystore (JKS) format. The resulting "keystore" is a repository for objects such as keys and certificates. The keystore is built using the command line Java keytool utility. This utility is available as part of the standard Java JDK Version 6 install. It is located in the *$JAVA_HOME*/bin directory.

For additional information about the keytool utility, refer to documentation located at the following URL:
http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html

Before you prepare your keystore, consider the following items:

- Where to store the keystore file (or files)
- What name and password to give your keystore
- Whether to use client-side authentication

NOTE The server always authenticates with the client. However, client-to-server (client-side) authentication is optional. Determine whether client-side authentication is required in your environment

Managing Keys and Certificates

You can use the `keytool` utility to create, store, and manage the keys and certificates you will need to use SSL with Service Activator. There are four basic steps you will need to carry out when preparing to use SSL with either the Operator UI or the Workflow Manager:

1. Generate a new key entry. A key entry consists of a public key certificate and a private key. Key entries are stored in the keystore. When a new key entry is generated, it is added to the keystore. If the keystore does not yet exist, it is created.
2. Generate a certificate request. This request is formatted to be submitted to a Certificate Authority (CA), such as VeriSign or Thawte.
3. Send the certificate request file to a Certificate Authority (CA), such as VeriSign or Thawte, for signing.
4. Import the signed certificate into your keystore.

The following section provides a generic example of how to complete these steps using the `keytool` utility. Specific instructions for Service Activator are provided beginning on page 74.

NOTE Be sure to read the generic example carefully, as it contains important details about using `keytool` that you need to understand before you generate the keys and certificates necessary for SSL to work with Service Activator.

Using the `keytool` Utility

1. Create a new key entry in the keystore named `my.keystore` with the password `mypass` using the following command:

   `keytool -genkey -keyalg RSA -alias <yourAliasName> -storepass \ mypass -keystore my.keystore`

   The `-alias` option specifies a shortened, keystore-specific name for an entity that has a key or certificate in the keystore. The `-keyalg` option specifies the algorithm that will be used to generate the key entry; use RSA with SSL.

   You will be prompted to fill in additional information including your name, organizational unit, organization, city or locality, state or province, and country. This information is used to create the distinguished name (DN) for your certificate. You will then be prompted for a key password. You can specify a password that is unique to your new key entry, or you can use the keystore password as your key password.

NOTE Only Step 1 is required to minimally configure a key and its associated certificate. Step 1 produces a self-signed certificate, which is less secure than a certificate signed by a CA. Steps 2 through 4 will replace the self-signed certificate with a certificate signed by a CA. In production environments, you are strongly encouraged to use certificates signed by a CA.

2. Generate a certificate request. In this case, the certificate request will be stored in the file named `my.csr`. You may specify any file name.

   `keytool -certreq -alias <yourAliasName> -file my.csr -keystore \ my.keystore`

   You will be prompted for both the keystore password and the key password. Once you supply these passwords, you should receive the following message:

   ```
   Certification request stored in file
   Submit this to your CA
   ```

3. Send the certificate request file (in this case, `my.csr`) to a Certificate Authority (CA), such as VeriSign or Thawte, for signing. Some CAs allow you to paste the contents of this file into an HTML form.

4. The CA will e-mail you a signed certificate. Save the certificate in a file. Import this file (in this case, `mysigned.cer`) into your keystore:

   `keytool -import -alias <yourAliasName> -file mysigned.cer -keystore \ my.keystore -trustcacerts`

   This import operation replaces the self-signed certificate associated with the alias *<yourAliasName>* with the signed certificate.

NOTE

If you use a nonstandard CA, you will need to import a CA root certificate as a trusted root certificate prior to importing your own certificates into the keystore. The Java SDK ships with the file `cacerts`, which contains the most common CA root certificates. The `-trustcacerts` option allows `keytool` to use those CA certificates. To import a CA certificate into the keystore, use the following command, where `ca.crt` is the file containing the root certificate for your CA:

`keytool -import -alias ca -file ca.crt -keystore my.keystore`

Configuring SSL for HTTPS (Operator UI)

There are three basic steps required to configure the Service Activator Operator UI to use SSL for HTTPS:

1. Load the server keystore.
2. Modify the JBoss `standalone.xml` file.
3. Start Service Activator.

Each of these steps will be described in detail in this section.

Step 1: Loading the Server Keystore (Operator UI)

This step includes creating the keystore, obtaining a signed certificate, and importing the signed certificate into the keystore.

a. Create a key entry in the keystore file named `activatorSSL.keystore` in the JBoss server configuration directory, *$JBOSS_HOME*`/server/default/conf`:

**_$JAVA_HOME_binkeytool -genkey -alias uialias -keyalg RSA **
-keystore _$JBOSS_HOME_serverdefaultconfactivatorSSL.keystore

The suggested alias, keystore name, and keystore location shown here are not mandatory. You may use any alias, name, and location you like. The keystore location and password, however, must match those values stored in the JBoss `standalone.xml` configuration file. See “Step 2: Modifying the JBoss Configuration Files” on page 74 for additional information.

b. Generate a certificate request, and store it in a file (in this case, `UIcert.csr`):

**keytool -certreq -alias uialias -file UIcert.csr -keystore **
_$JBOSS_HOME_serverdefaultconfactivatorSSL.keystore

c. Submit your certificate request to a Certificate Authority, such as VeriSign or Thawte.

d. Upon receiving your signed certificate, save it in a file (in this case `UIsigned.cer`), and import it into your keystore:

**keytool -import -alias uialias -file UIsigned.cer -keystore **
_$JBOSS_HOME_serverdefaultconfactivatorSSL.keystore -trustcacerts

Remember to use the same passwords in the `-import` operation that you used when you generated the key entry.

CAUTION

Be sure to check a certificate very carefully before importing it as a trusted certificate.

Step 2: Modifying the JBoss Configuration Files

Once you have configured JSSE and loaded your certificates, you must configure JBoss to take advantage of the SSL functionality. To do this, modify the JBoss `standalone.xml` file to add an HTTPS connector. This file is located in the following directory:

$JBOSS_HOME`/standalone/configuration`

Add the following HTTPS Connector to the subsystem urn:jboss:domain:web:1.0:

```
<subsystem xmlns="urn:jboss:domain:web:1.0"
default-virtual-server="default-host">

  <connector name="http" protocol="HTTP/1.1" socket-binding="http"
scheme="http"/>

  <connector name="https" protocol="HTTP/1.1" socket-binding="https"
scheme="https" secure="true">

    <ssl name="keyalias" password="verySecret"
certificate-key-file="/opt/HP/jboss/my.keystore" protocol="ANY"
verify-client="false"/>

  </connector>

  <virtual-server name="default-host" enable-welcome-root="true">

  <alias name="localhost"/>

  <alias name="example.com"/>

  </virtual-server>

</subsystem>
```

Set `certificate-key-file` to the location and name you selected for your keystore, and set `password` to match your keystore password. If you want to use client-side authentication, set `verify-client` to "`true`."

Configuring the JBoss Operator UI Port

In the JBoss `standalone.xml` file, the `port` attribute is defined. By default, the `port` attribute for `HTTPS` is 8443. This attribute is the TCP/IP port number on which JBoss will listen for secure connections. You can change this to any port number you wish (such as the default port for `HTTPS` communications, which is 443).

Step 3: Starting Service Activator

You will need to restart Service Activator to have your configuration changes take effect. To do this, follow the instructions in "Starting and Stopping Service Activator" on page 42 of the *HP OpenView Service Activator—Installation Guide*.

Configuring SSL for Secure Message Transmission (Workflow Manager)

There are three basic steps required to configure the Service Activator Workflow Manager to send and receive secure messages using SSL:

1. Load the server keystore.
2. Modify the Workflow Manager configuration file.
3. Restart the Workflow Manager.

Each of these steps will be described in detail in this section.

Step 1: Loading the Server Keystore (Workflow Manager)

This step includes creating the keystore, obtaining a signed certificate, and importing the signed certificate into the keystore.

a. Create a key entry in the keystore file named `mwfmSSL.keystore` in the *`$ACTIVATOR_ETC`*`config` directory:

```
$JAVA_HOMEbinkeytool -genkey -alias mwfmalias -keyalg RSA \
-keystore $ACTIVATOR_ETCconfigmwfmSSL.keystore
```

The suggested alias, keystore name, and keystore location shown here are not mandatory. You may use any alias, name, and location you like.

b. Generate a certificate request, and store it in a file (in this case, `mwfmcert.csr`):

```
keytool -certreq -alias mwfmalias -file mfwmcert.csr -keystore \
$ACTIVATOR_ETCconfigmwfmSSL.keystore
```

c. Submit your certificate request to a Certificate Authority, such as VeriSign or Thawte.

d. Upon receiving your signed certificate, save it in a file (in this case, `mwfmsigned.cer`), and import it into your keystore:

```
keytool -import -alias mwfmalias -file mfwmsigned.cer \
-keystore $ACTIVATOR_ETCconfigmwfmSSL.keystore -trustcacerts
```

Step 2: Modifying the Workflow Manager Configuration File

Change the values of the `keystore` and `keystore_password` parameters in the SocketListenerModule and SocketSenderModule specifications in the `mwfm.xml` file to match the keystore name and password, respectively, that you select. Also change the value of the `clientauth` parameter for the SocketListenerModule to reflect the type of authentication you will use. See Chapter 5, “Configuring the Workflow Manager,” on page 343 of *HP OpenView Service Activator—Workflows and the Micro-Workflow Manager* for additional information about editing this file.

Step 3: Restarting the HP Service Activator

You will need to stop and restart the Workflow Manager to have your configuration changes take effect. To do this, follow the instructions in the *HP OpenView Service Activator—Installation Guide*.

Troubleshooting

Many things can go wrong when working with JSSE and certificates. Here is a list of common problems and their solutions:

java.security.NoSuchAlgorithmException: Algorithm SunX509 not available or java.security.NoSuchAlgorithmException: Algorithm TLS not available

This common error indicates that you did not specify your security algorithm providers properly. If you configured the algorithms by modifying the `java.security` file, check to be sure that you modified the correct file and that you are executing the correct `java.exe`. Run Java with the `-version` flag to check the version number of the Java SDK you are currently using.

If the version of your SDK is correct, check the `java.security` file carefully to be sure that your `security.provider.#` line is not being overridden by another `security.provider.#` line later in the file. Next, be sure that the order of `security.provider.#` lines is sequential from 1 to #. The security manager will not recognize any provider settings if there is a gap in the number sequence.

javax.net.ssl.SSLException: untrusted server cert chain or javax.net.ssl.SSLException: Received fatal alert: certificate_unknown

These exceptions will be thrown if a server or client is unable to validate the credentials provided by the other party. For instance, if a certificate is not signed by any other certificates known (and trusted) by the trust manager, the certificate will be rejected. If you are having this problem with two parties that should be trusting each other, verify that each certificate has been imported into the keystore of the other and that the certificate authority used to sign each certificate has been distributed properly.

java.io.IOException: Keystore was tampered with, or password was incorrect

This error typically indicates that the password provided to retrieve the certificates from the local keystore is incorrect, but it could also mean that something is wrong with the keystore file itself. The file might be corrupted, or the file permissions might be too restrictive.

javax.net.ssl.SSLException: No available certificate corresponds to the SSL cipher suites which are enabled.

This exception is typically thrown when a connection is being initialized. It means that a socket or server socket object does not have any certificates, or not the right kind of certificates, to use when starting communication or listening on the port. To solve this, make sure that the keystore file is being loaded correctly, that it is the keystore you intended to use, and that the context is initialized with the right set of key and trust managers.

Client Hangs While Connecting

The client may hang if it is trying to use a cleartext socket, but the server is using TLS. Since the server is expecting a stream containing protocol negotiation data, it will wait on the open socket until it hears what it is listening for. Eventually the client will time out.

Finding Additional Information

If you experience a problem with your SSL implementation that is not addressed by one of the solutions discussed in this section, examine the following log files for further information:

Component	Log Files To Examine
Operator UI or JBoss	`boot.log` `server.log.*`
Workflow Manager	`mwfm_active.log.xml`

F Quick Installation Guide

This appendix is a quick installation guide for a Windows installation. In this guide, you can find the hardware and software requirements for installing Service Activator as well as the detailed instructions to guide you through installing the product. The document is intended for persons giving demonstrations of Service Activator. The product is installed with the default settings. For any instructions about adjusting the settings, please refer to the relevant chapters in this guide.

As examples, this installation guide uses the database instance **sa**, the database user name **ovactusr**, and the user password **abcde**.

Hardware and Software Requirements

Hardware

- 256 MB of memory
- 40 MB of disk space available on the boot drive
- 150 MB of disk space for Service Activator
- Minimum of 400 MB in virtual memory
- Disk space for Oracle Server

Software

Before installing Service Activator, install and configure the following software on the activation server.

IMPORTANT The software must be installed in the order listed.

1. Microsoft Windows 2003 or XP
2. Java 2 Software Developers Kit (SDK) 1.4.2_06
3. Open SSH Secure Shell (SSH) 3.6 or later (optional, see “Step 3: Install and Configure Secure Shell (SSH)” on page 84)
4. Microsoft Internet Explorer 6.0 or Netscape Navigator 7.1 (or later versions of either browser)
5. Oracle (on a single server)

Once the PCs are set up as described above, Service Activator can be installed.

Installation steps

Step 0: Get Required Software

Download the Cygwin from *http://www.cygwin.com*. Make sure you download these packages: `Admin -> cygrunsrv, Base -> All, Net -> openssh`

Step 1: Install Microsoft Windows 2003 or XP

This is a straightforward operation. Install patches and service packs.

Step 2: Install Java Software Development Kit (SDK) 1.4.2

1. Download the software from *http://java.sun.com/products/archive/index.html*
2. Install the software under `c:`
3. Set `JAVA_HOME` System Environment variable to the SDK install location
4. Add `$JAVA_HOME/bin` to the system `PATH`

Step 3: Install and Configure Secure Shell (SSH)

Follow these steps to create the necessary encryption keys for `ovactusr`:

1. Log on to your system as `ovactusr`. Make sure you choose your local computer in the box labelled Log on to:, and not a domain on your network. Your local computer is usually marked (this computer) in the drop-down list.
2. Open a command line window `(cmd.exe)`, and change directories to the `C:\cygwin` directory. Type **`cygwin`** and press `Enter`. This will put you in the Cygwin command prompt mode. This is a UNIX emulation environment so all commands are in UNIX format (case-sensitive and with '/' as the file-separator character).
3. Type **`ssh-user-config -y -p "`**. You should see the following text in the command shell window:

   ```
   $ ssh-user-confg -y -p ""
   Generating /home/ovactusr/.ssh/identity
   Adding to /home/ovactusr/.ssh/authorized_keys
   Generating /home/ovactusr/.ssh/id_rsa
   Adding to /home/ovactusr/.ssh/authorized_keys
   Generating /home/ovactusr/.ssh/id_dsa
   Adding to /home/ovactusr/.ssh/authorized_keys
   ```

4. Allow the `SYSTEM` user to read and write the private keys belonging to `ovactusr`. This is necessary because the Service Activator Resource Manager runs as a service. Type the following commands:

   ```
   cd/home/ovactusr/.ssh
   setfacl -m user: SYSTEM:rw- id_dsa id_rsa identity
   ```

   You have just created an encryption key pair for `ovactusr` on you Service Activator server. Next, configure Secure Shell on your target machines. After that, test the connections between your Service Activator server and your targets.

5. Once you have completed the above steps, verify that the SSH server and client are installed properly:
   - type **`ssh localhost`** from a command line prompt
   - when this connection is initiated, reply **`Yes`** when prompted to accept the host key

Step 6: Install Internet Explorer or Netscape Navigator

Install Internet Explorer 6.0 or Netscape Navigator 7.0. You may choose to install later versions of either browser.

Step 7: Oracle Installation

Oracle 8.1.7 or 9.2 must be installed on your server.

1. Create a database instance named **`sa`** for use by Service Activator (100 MB minimum is recommended).
2. Create a database user (for example, **`ovactusr`**), set the user password (for example, **`abcde`**).
3. Grant the user dba permissions.

Step 8: Install Service Activator

1. Log on to your PC as an administrator user.
2. Start the Service Activator Install shield. Press `Next`.
3. Confirm that you accept the Licence Agreement by pressing `Yes` in the Licence Agreement window.
4. In the `User Information` window, enter the user name and the company name (for example, User: **`ovactusr`** Company: **`company`**) and press `Next`.
5. In the next window, select the installation drive and press `Next`.
6. Then, the `Installing` window will appear showing the current settings. Press `Next` to continue installation.
7. Press `Next` in the `Configuring Service Activator` window to proceed.
8. Press `OK` in the `HP Service Activator Installation` message.
9. The `General` window will appear showing the default port numbers for the Micro-Workflow Manager (port no. 2000), the Resource Manager (6667), the Oracle Database Listener Port (1521) and the Web Server port (8080). Press `Next` to accept the default values.
10. In the next window, complete the Oracle database configuration details.
    - Fill in the database user name. Use the user name created in Step 7: Oracle Installation
    - Fill in and confirm the user password. Use the password created in Step 7: Oracle Installation.
    - Fill in the system name for the Oracle server.
    - Fill in the name of the database. Use the database name created in Step 7: Oracle Installation (**`sa`**).

- Check the `create database` check-box.
- From the drop-down list, select **`All parts of Service Activator`** to indicate how the system will use the database.
- Press `Next`.

11. In the next window, complete the SSH configuration details.
    - Set the user name to **`ovactusr`** (created during SSH setup).
    - Locate the identity file: `C:\cygwin\home\ovactusr\.ssh\identity`.
    - Type **`sudo`** in the `Unix sudo program` field.
    - Type **`C:\cygwin\bin`** in the next field.
    - Press `Next`.
12. Wait for the `AutoPass:Licence Management` window to open.
    - From the available options in the left-hand pane, click on the `Install/Restore Licence Key from File`.
    - Place the cursor in the `File Path` field and browse to the file containing licence information.
    - Once you have selected the licence file, press `View File Contents`. This will display the list of licences to be installed.
    - Select the licences to install and press `Install`. Then, close the `AutoPass:Licence Management` window.
    - You will be returned to the Install Shield window. Press `Finish` to complete installing Service Activator.

To verify the Service Activator installation, follow these steps:

1. Start the services
    - Use the Windows Service Manager to make sure the following services are not running: `HPOVACT_JBOSS`, `HPOVACT_MWFM`, and `HPOVACT_RESMGR`.
    - Restart each of them in the above order.
2. Browse to the login screen
    - In the address line of your Internet browser, enter: *http://<machine_name>/activator/login.html*
    - This should bring up the login screen for Service Activator.
3. Log in to Service Activator
    - Type any text into the username field and press `Enter`.
    - This should bring up a new page that is not an error message.