HP Service Manager

Software Version: 9.41

For the supported Windows® and UNIX® operating systems

System Installation and Setup help topics for printing

Document Release Date: September 2015
Software Release Date: September 2015

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

For a complete list of open source and third party acknowledgements, visit the HP Software Support Online web site and search for the product manual called HP Service Manager Open Source and Third Party License Agreements.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hp.com/.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HP Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support site at: https://softwaresupport.hp.com.

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

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

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HP Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: https://softwaresupport.hp.com/web/softwaresupport/access-levels.

HPSW Solutions Catalog accesses the HPSW Integrations and Solutions Catalog portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01702710.

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not

be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Configuring installation and setup options

HP Service Manager system installation and setup includes a number of implementation options you can configure after installation. The following table lists the options you can enable or configure from a new development environment installation.

Feature	Description	Default state in new installations
Computer Telephony Integration (CTI) with the Web client	An integration option allowing Web clients to access telephony software	Disabled
Encryption of configuration file settings	A security option that protects values listed in the configuration file	Disabled
Encryption of operator passwords	A security option that protects the passwords listed in operator records	**Enabled**
Event Services	A group of applications that allow Service Manager to send data to and to receive data from external systems	Disabled
"Lightweight Directory Access Protocol (LDAP)" on page 86	An integration option that allows Service Manager to access data stored in external directory services	Disabled
"Client-side printing" on page 102	An implementation option that allows clients to print to their local printer	**Enabled**
"Servlet implementation" on page 106		
• "Horizontal scaling implementation" on page 127	An implementation option that uses servlet implementation features to manage Service Manager instances running on multiple hosts	Disabled
• "Single servlet implementation" on page 108	An implementation option where a single servlet container process manages all client connections	**Enabled**
• "Requirements for a vertical scaling and required SSL implementation" on page 125	A vertical scaling implementation option running in combination with the "Required SSL" security option	Disabled
• "Requirements for a vertical scaling and required SSL implementation" on	A implementation option where two or more servlet container processes manage all client connections on a single host	Disabled

Feature	Description	Default state in new installations
page 125		
"UTF-8 conversion" on page 189		
• Background UTF-8 conversion	An implementation option where a background process runs over time to convert all Service Manager data to UTF-8 format	Disabled
• On demand UTF-8 conversion	An implementation option where the server converts a single record to UTF-8 format the first time a user accesses the record	**Enabled**
Web Services		
• External Web Services	An integration option that allows Service Manager to send data to and to receive data from external Web Services	**Enabled**
• Publish Service Manager applications as Web Services	An integration option that allows Web Services clients to access Service Manager applications	Disabled

HP Service Manager Interactive Installation Guide

The *HP Service Manager Interactive Installation Guide* aids Service Manager implementers who are responsible for installing and configuring Service Manager. The guide also has instructions on the following topics:

- How to install the Service Manager server and client on Windows and Unix platforms
- How to deploy HP Service Manager on a web application server
- How to install the help on a web server or local file system
- How to install the Open Database Connectivity (ODBC) driver
- How to start the legacy listener
- How to install and configure the client customization utility

The *HP Service Manager Interactive Installation Guide* is available from the installation media or you can download it from the HP Software Support Online web site.

Upgrade Assessment Toolkit User Guide

The *Upgrade Assessment Toolkit User Guide* aids HP partners in estimating the amount of effort required for a data migration from ServiceCenter to Service Manager or a system upgrade from a lower version of Service Manager to a higher one. This guide provides instructions on how to generate and customize data reports using a toolkit named Assessment Toolkit, and also describes a suite of assessment reports that can be generated with this toolkit. With the help of these assessment reports, HP partners can easily estimate migration effort and identify potential data migration problems.

The Assessment Toolkit is critical for successful migration from ServiceCenter to Service Manger and system upgrade from a lower version of Service Manger to a higher one. Before a system migration or upgrade, HP partners can generate assessment reports to get an overview of the differences between the related ServiceCenter data, the Service Manager out-of-box (OOB) data, and the customized data from a real production environment that needs to be migrated as well. Consequently, HP partners are able to decide what data can or cannot be migrated and how it will be migrated.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Upgrade Assessment Toolkit User Guide* is available from the help.

HP Service Manager Upgrade Guide

The *HP Service Manager Upgrade Guide* aids Service Manager implementers who are responsible for upgrading the Service Manager to the latest version. The guide has instructions on how to upgrade the Service Manager client, server, and applications. Implementers can use this guide to complete an application upgrade or to run their current application versions on an upgraded client/server environment.

HP provides two options for viewing the *HP Service Manager Upgrade Guide*:

1. You can view an interactive version of the upgrade guide. The *HP Service Manager Interactive Upgrade Guide* enables you to select the options required for your implementation and produce a customized upgrade guide that includes only the requirements and tasks that apply to you.

2. You can view the PDF versions of the upgrade guides that are embedded in the *HP Service Manager Installation and Upgrade Documentation Center*.

Both the *HP Service Manager Interactive Upgrade Guide*and the *HP Service Manager Installation and Upgrade Documentation Center* are available on the installation media or you can download it from the HP Software Support web site.

HP may update product documentation after initial release. The latest version of the *HP Service Manager Upgrade Guide* is available from the HP Software Support web site.

HP Service Manager Applications Patch Manager Guide

The *HP Service Manager Applications Patch Manager Guide* aids Service Manager implementers who are responsible for updating the Service Manager applications to the latest patch release.

This guide is available from the HP Software Manuals Site at:
http://h20230.www2.hp.com/selfsolve/manuals

Autopass licensing changes

All HP Service Manager installations now include a bundled version of Autopass that no longer requires a separate installation of the Autopass software. The bundled version of Autopass is specifically for verifying the Service Manager license and as such requires you to place the LicFile.txt file in a new path. Your LicFile.txt file should now reside in your server's RUN folder. See the *HP Service Manager Installation Guide* for information about Autopass settings and options.

Server implementation options

HP Service Manager supports several different implementation options to manage client connections to the server. The following is a list of the most common server implementation options and the features they offer.

Feature	Single servlet implementation	Vertical scaling implementation	Vertical scaling with SSL implementation	Horizontal scaling implementation
Specify all communication ports used?	Yes	Yes	Yes	Yes
Specify connection limit?	Yes	Yes	Yes	Yes
Load balances client connections?	No	Yes	Yes	Yes
Implementation uses multiple hosts?	No	No	No	Yes
Allows dynamic addition and removal of hosts to virtual group?	No	No	No	Yes
Allows dynamic addition and removal of servlet container processes to virtual group?	No	Yes	Yes	Yes
Options available to convert current implementation into new one	• Vertical scaling • Horizontal scaling	Horizontal scaling	• Vertical scaling • Horizontal scaling	Not applicable

Hardware load balancers

Horizontal or vertical scaling implementations can now use a supported third-party hardware load balancer in place of a Service Manager load balancer process. Hardware load balancers can offer the following advantages over a Service Manager load balancer process:

- Offer a variety of load balancing algorithms
- Offer SSL acceleration or offloading capabilities
- Offer error detection and correction functionality
- Increase system resources available on a Service Manager host by removing a load balancer process
- Reduce the number communications ports required between Service Manager clients and servers
- Offers option to run Service Manager from a private network with the hardware load balancer acting as a network gateway

Note: Some customers may choose to omit the web server component in their implementations. While this is a viable configuration (the exact reasons for choosing such a configuration vary), HP recommends that customers adhere to the best practice of implementing the Service Manager web tier product in a standard web server with Java application server configuration. That is, you should use a combination of products, such as an Apache HTTP Server and Apache Tomcat to host the Service Manager web tier. This is particularly important when there is a requirement to implement SSO or TSO functionality in your environment.

Configure an F5 hardware load balancer

The following topics describe how to configure an F5 HWLB to work with Service Manager.

Configure an HWLB to accept client requests in HTTP mode

Note: This topic applies to Windows clients and to HP Service Manager web tier clients.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- The F5 management URL can be accessed through the management IP.
- A VLAN is created on the interfaces.

- Static IPs are available for the interfaces.

Step 1: Configure Secure Network Address Translation (SNAT)

If the server and client are located in the same local network, SNAT is required. To determine whether you must configure SNAT, please consult your Network Engineering team.

To configure SNAT, follow these steps:

1. Click **Local Traffic** > **SNATs** > **SNAT Pool List**.
2. Click **Create** and then type `TEST_SNAT` in the **Name** field.
3. Type the IP address of the Virtual Server IP in the **IP Address** field, and then click **Add**.
4. Click **Finished**.

Step 2: Configure an iRule to enable session persistence

Configure an iRule to enable session persistence for HTTP requests, based on the JSESSIONID cookie that the Service Manager server provides. To do this, follow these steps:

1. Click **Local Traffic** > **iRules** > **iRule List**.
2. Click **Create**, and then type `TEST_JSESSION_IRULE` in the **Name** field.
3. Copy the following text into the **Definition** area:

```
when HTTP_REQUEST {
if {[HTTP::cookie "JSESSIONID"] ne ""}{
persist uie [string tolower [HTTP::cookie "JSESSIONID"]] 300
} else {
set jsess [findstr [string tolower [HTTP::path]] "jsessionid=" 11 ";"]
if { $jsess != "" } {
persist uie $jsess 300
}
}
}when HTTP_RESPONSE {
if {[HTTP::cookie "JSESSIONID"] ne ""} {
persist add uie [string tolower [HTTP::cookie "JSESSIONID"]] 300
}
```

```
}
```

4. Click **Finished**.

Step 3: Disable OneConnect transformations

If OneConnect transformations are enabled in the default HTTP profile, you must create a new HTTP profile.

Note: Only complete this step on HWLB configurations that use HTTP mode to connect to a web service (such as SRC).

To do this, follow these steps:

1. Click **Local Traffic** > **Virtual Servers** > **Profiles** > **Services** > **HTTP**.
2. Click Create, and then type the TEST_HTTP in the Name field.
3. Click the **Custom** check box.
4. Disable the OneConnect Transformations, and then click **Finish**.

Step 4: Create a new TCP-based health monitor

A TCP-based health monitor detects whether the Service Manager process is already listening to the port. You can also use the interface provided by Service Manager to check the RTE servlet status. For more information, see "Configure an HWLB Health Monitor for the Service Manager server" on page 24.

To create a new TCP-based health monitor, follow these steps:

1. Click **Local Traffic** > **Monitors**.
2. Click **Create**, and then type `TEST_TCP` in the **Name** field.
3. Select **TCP** in the **Type** field, and then click **Finished**.

Step 5: Create a node

To create a node, follow these steps:

1. Click **Local Traffic** > **Nodes** > **Node List**.
2. Click **Create**, and then type the SM server host IP address.

Note:

- There may be many Service Manager nodes (for example, in a horizontally-scaled Service Manager system).
- You may add as many nodes as are required by your deployment.

Step 6: Configure a pool member

To configure a pool member, follow these steps:

1. Click **Local Traffic** > **Pools** > **Pool List**.
2. Click **Create**, and then type `TEST_POOL` in the **Name** field.
3. Select **TEST_TCP** as the health monitor.
4. Click **Node List** for the New Members field.
5. Select the node you prepared in step 5 and then click **Add**.
   Repeat this step to add all the nodes you want to put in the pool.
6. Click **Finished**.
7. In the pool list, click the corresponding number in the Members column.
8. On the **Members** tab, verify the members.
   Select the members that you want to enable and then click **Enable**.

Step 7: Create a new virtual server

To create a new virtual server, follow these steps:

1. Click **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Click **Create**, and then type `TEST_VS` in the **Name** field.

3. Specify a virtual server IP address and port number.
4. Select **oneconnect** as the OneConnect profile.
5. Select **http** or **TEST_HTTP** as the **HTTP Profile** option.
6. Select **TEST_SNAT** as the **SNAT Pools** option.
7. Enable the **TEST_JSESSION_IRULE** iRule that you created previously.
8. Select **TEST_POOL** as the **Default Pool** option.
9. Click **Finished**.

Configure an HWLB for SSL offloading

You must implement the F5 HWLB's SSL offloading feature to ensure that the load balancer can correctly inspect HTTPS traffic. This is required in order to maintain session persistence.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- The HWLB is configured to accept client requests in HTTP Mode. For more information about how to do this, see "Configure an HWLB to accept client requests in HTTP mode" on page 17.
- The required Service Manager certificates are generated. For more information about how to do this, see the following article in the HP knowledge base:

  https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM00488034 (Generating SSL Profiles for an F5 Hardware Load Balancer on Service Manager 9.32 - Supporting Files)

Configure the virtual server to use the SSL profile (client)

To configure the virtual server to use the SSL profile for the client, follow these steps:

1. Click **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Select the virtual server that you use to load balance client requests to Service Manager servlets.

3. In the **SSL Profile (Client)** drop-down list, select **TEST_SSL_CLIENT**, and then click **Update**.

Note: You can use the JSession Persistence iRule to maintain session persistence.

Configure an HWLB for SSL between an F5 load balancer and a Service Manager server

Prerequisites

Before you begin this process, verify that the following conditions are true:

- The HWLB is configured for SSL offloading. For more information about how to do this, see "Configure an HWLB for SSL offloading" on the previous page.
- The required Service Manager certificates are generated and the SSL Profiles are prepared. For more information about how to do this, refer to Generating SSL Profiles for an F5 Hardware Load Balancer on Service Manager in the HP Knowledge Base.

Step 1: Create a pool that contains HTTPS ports as its pool members

To create a pool that contains HTTPS ports, follow these steps:

1. Click **Local Traffic** > **Pools** > **Pool list**.
2. Click **Create**, and then type `TEST_SSL_SERVER` in the **Name** field.
3. Select a health monitor. For more information, see "Configure an HWLB Health Monitor for the Service Manager server" on page 24.
4. In the **address** field, type the IP address of the Service Manager server, and then set the server port to the HTTPS port of the Service Manager server.
5. Click **Add**.
6. Repeat step 4 and 5 to add all the HTTPS ports, and then click **Finished**.

Note: You can add as many pool list items as your deployment requires.

Step 2: Configure the session persistence type

To configure the session persistence type, follow these steps:

1. Click **Local Traffic** > **Profiles** > **Persistence**.
2. Click **Create**.
3. In the **Name** field, type `TEST_COOKIE_INSERT`.
4. In the **Persistence Type** drop-down list, select **Cookie**.
5. Click to select the **Custom** option on the right-hand side.
6. In the **Cookie Method** drop-down list, select **HTTP Cookie Insert**.
7. Click **Finish**.

Step 3: Configure the virtual server to use the SSL profile (Server)

To enable the virtual server to use the server SSL profile, follow these steps:

1. Click **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Select the virtual server that you use to load balance client requests to Service Manager servlets.
3. Select **oneconnect** as the OneConnect profile.
4. In the **SSL Profile (Client)** drop-down list, select **TEST_SSL_CLIENT**.
5. In the **SSL Profile (Server)** drop-down list, select **TEST_SSL_SERVER**.
6. Set the SNAT Pool to **TEST_SNAT**, and then click **Update**.
7. On the **Resource** tab, select the default Pool that contains the HTTPS port of the Service Manager server.
8. Set the Default Persistence Profile to **TEST_COOKIE_INSERT**.

Configure an HWLB Health Monitor for the Service Manager server

The health monitor helps the HWLB to identify whether a Service Manager servlet can accept a new connection. This prevents the servlet returning errors if the HWLB directs a request to the servlet when the servlet is at full capacity or when Service Manager is running in Quiesce mode.

This section describes how to configure and use a health monitor with an F5 HWLB.

Prerequisites

Before you begin this process, verify that the HWLB is configured to accept client requests in HTTPS mode (if you use an SSL port) or in HTTP mode (if you use a non-SSL port). For more information about how to do this, see "Configure an HWLB to accept client requests in HTTP mode" on page 17.

Step 1: Create the health monitor on the F5 HWLB

To create the health monitor on the F5 HWLB, follow these steps:

1. Click **Local Traffic** > **Monitors** > **Create**.
2. Type a name for the health monitor, such as `TEST_MONITOR`.
3. Select **HTTP** or **HTTPS** as the type, depending on the connection type between the F5 HWLB and the Service Manager server.
4. In the **Send String** field, type `GET /mbeanclient-X.XX/Action?getattribute=AcceptNewConnection \r\n`.
5. In the **Receive String** field, type `true`.
6. If you use an SSL connection between the F5 HWLB and the Service Manager server, select **TEST_SSL_SERVER** in the **Client Certificate** and **Client Key** fields.
7. Click **Finished**.

Step 2: Update the pool list

1. Click **Pools** > **Pool List**.
2. Select the pool that your virtual server uses.

3. In the **Health Monitors** section, move **TEST_MONITOR** from the **Available** side to the **Active** side.
4. Click **Update**.

Step 3: Modify the sm.ini file and restart Service Manager

1. Add the following line to the `sm.ini` file:

   ```
   JVMOption999:../../lib/mbeanclient- X.XX.war
   ```

2. Restart the Service Manager servers.

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in HTTP mode

Access Service Manager on a Windows client through an F5 HWLB in HTTP mode

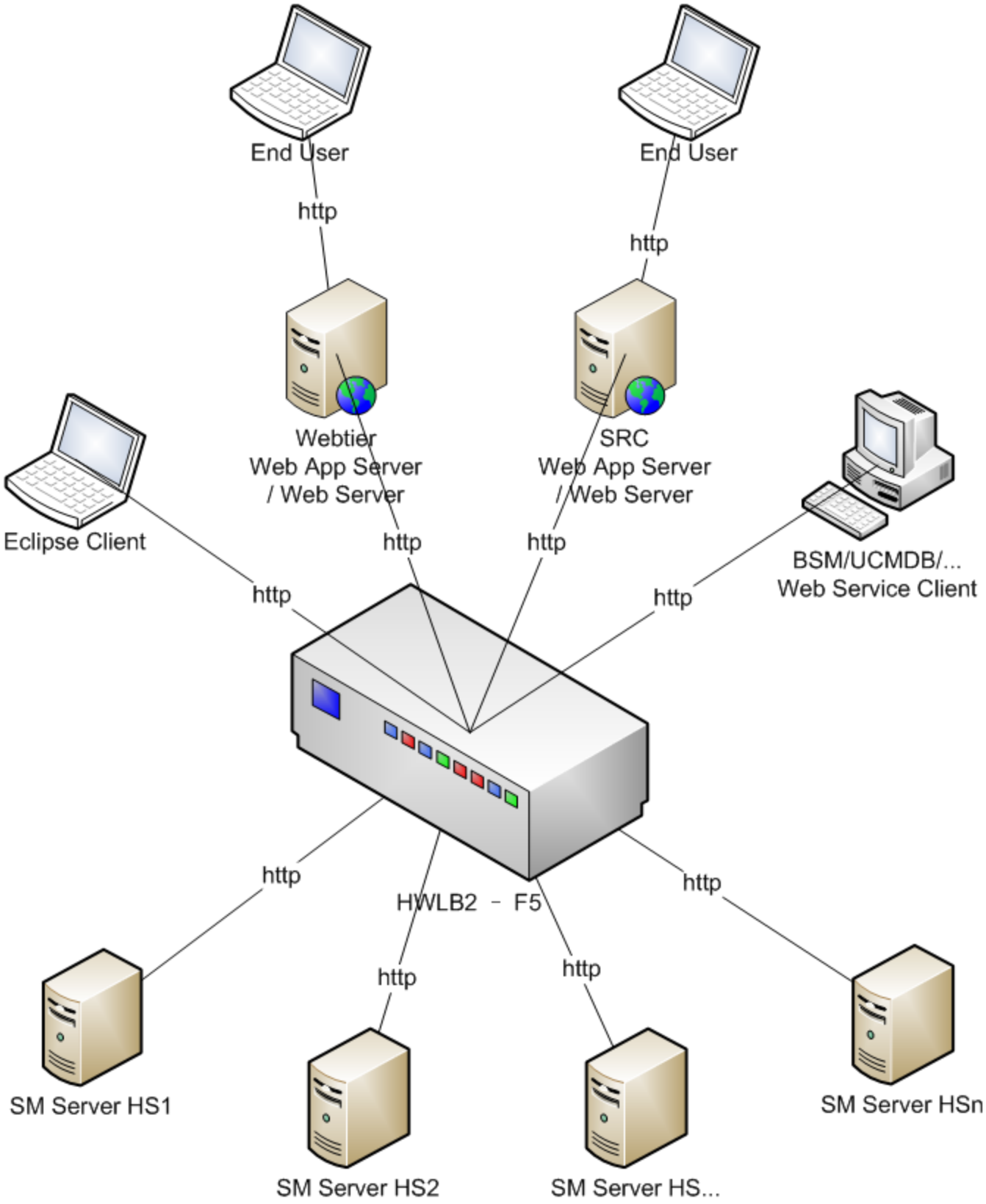
Access Service Manager on a web client through an F5 HWLB in HTTP mode

Common F5 hardware load balancer deployments

The following topics describe some common F5 HWLB deployments.

Access Service Manager through an F5 HWLB in HTTP mode

The following figure illustrates a deployment in which users access Service Manager through an F5 HWLB in HTTP mode:


End User
End User
http
http
Webtier
Web App Server
/ Web Server
SRC
Web App Server
/ Web Server
Eclipse Client
BSM/UCMDB/...
Web Service Client
http
http
http
http
HWLB2 - F5
http
http
http
http
SM Server HS1
SM Server HS2
SM Server HS...
SM Server HSn

Access Service Manager on a Windows client through an F5 HWLB in HTTP mode

This deployment is suited to testing HP Service Request Catalog and Service Manager in a development or non-production environment. As you can see from the figure in "Access Service Manager through an F5 HWLB in HTTP mode" on page 25, this configuration does not use HTTPS/SSL encrypted communication. This deployment should be used for sand box, development, or demonstration (proof of concept) purposes.

This section describes how to access Service Manager on a Windows client through an F5 HWLB in HTTP mode.

Step 1: Configure a Windows client to support an external hardware load balancer

To configure a Windows client to support an external hardware load balancer, follow these steps:

1. Install a new Windows client.
2. Start the Windows client, and then create a new network connection called `hwlb_connect`.
3. Type the IP or host name of the virtual server as the connection's server address.
4. Type the port of the HWLB virtual server as the connection's server port.
5. On the **Advanced** tab, click to select the **Connect to External Load Balancer** option.
6. Click **Apply**, and then click **Connect**.

Step 2: Enable an HWLB on a Service Manager server

To enable an HWLB on a Service Manager server, follow these steps:

1. Add the `external_lb` attribute to the `sm.ini` file on the Service Manager server.

   Note: The `external_lb` attribute is a system attribute, and every servlet node on the same host should use the same attribute setting. In most cases, this attribute needs to be configured in the `sm.ini` file when you start a servlet from the operating system command prompt.

However, to prevent a servlet node from running in external load balancer mode, you can include `-external_lb:0` in a command when you start a servlet from the operating system command prompt. As the servlet node does not work in external load balancer mode, it rejects any connections from a hardware load balancer.

Only `-external_lb:0` can be added to a command. `-external_lb:1` and `external_lb` are not recognized in commands.

2. Make sure that no Service Manager software load balancer (for example, `sm -loadBalancer`) is configured or running on the Service Manager server.

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in HTTP mode

Access Service Manager on a web client through an F5 HWLB in HTTP mode

F5 HWLB Health Monitor

Access Service Manager on a web client through an F5 HWLB in HTTP mode

This deployment is suited to testing HP Service Request Catalog and Service Manager in a development or non-production environment. As you can see from the figure in "Access Service Manager through an F5 HWLB in HTTP mode" on page 25, this configuration does not use HTTPS/SSL encrypted communication. This deployment should be used for sand box, development, or demonstration (proof of concept) purposes.

This section describes how to access Service Manager on a web client through an F5 HWLB in HTTP mode.

Step 1: Configure a web client to support an external load balancer

To configure a web client to support an external load balancer, follow these steps:

1. Set the `externalLB` parameter in the `web.xml` file to `true`, as follows:

```
<init-param>
      <param-name>externalLB</param-name>
      <param-value>true</param-value>
```

```
 </init-param>
```

2. Set the host and port to match the HWLB virtual server IP and port. To do this, configure the `web.xml` file as follows:

```
<init-param>
      <param-name>serverHost</param-name>
      <param-value><fully-qualified domain name of the HWLB virtual server></param-value>
 </init-param>
 <init-param>
      <param-name>serverPort</param-name>
      <param-value><Port of the HWLB virtual server></param-value>
 </init-param>
```

Step 2: Enable an HWLB on a Service Manager server

To enable an HWLB on a Service Manager server, follow these steps:

1. Add the `external_lb` attribute to the `sm.ini` file on the Service Manager server.

   Note: The `external_lb` attribute is a system attribute, and every servlet node on the same host should use the same attribute setting. In most cases, this attribute needs to be configured in the `sm.ini` file when you start a servlet from the operating system command prompt.

   However, to prevent a servlet node from running in external load balancer mode, you can include `-external_lb:0` in a command when you start a servlet from the operating system command prompt. As the servlet node does not work in external load balancer mode, it rejects any connections from a hardware load balancer.

   Only `-external_lb:0` can be added to a command. `-external_lb:1` and `external_lb` are not recognized in commands.

2. Make sure that no Service Manager software load balancer (for example, `sm -loadBalancer`) is configured or running on the Service Managerserver.

Click here to show or hide links to related topics.

Related Tasks

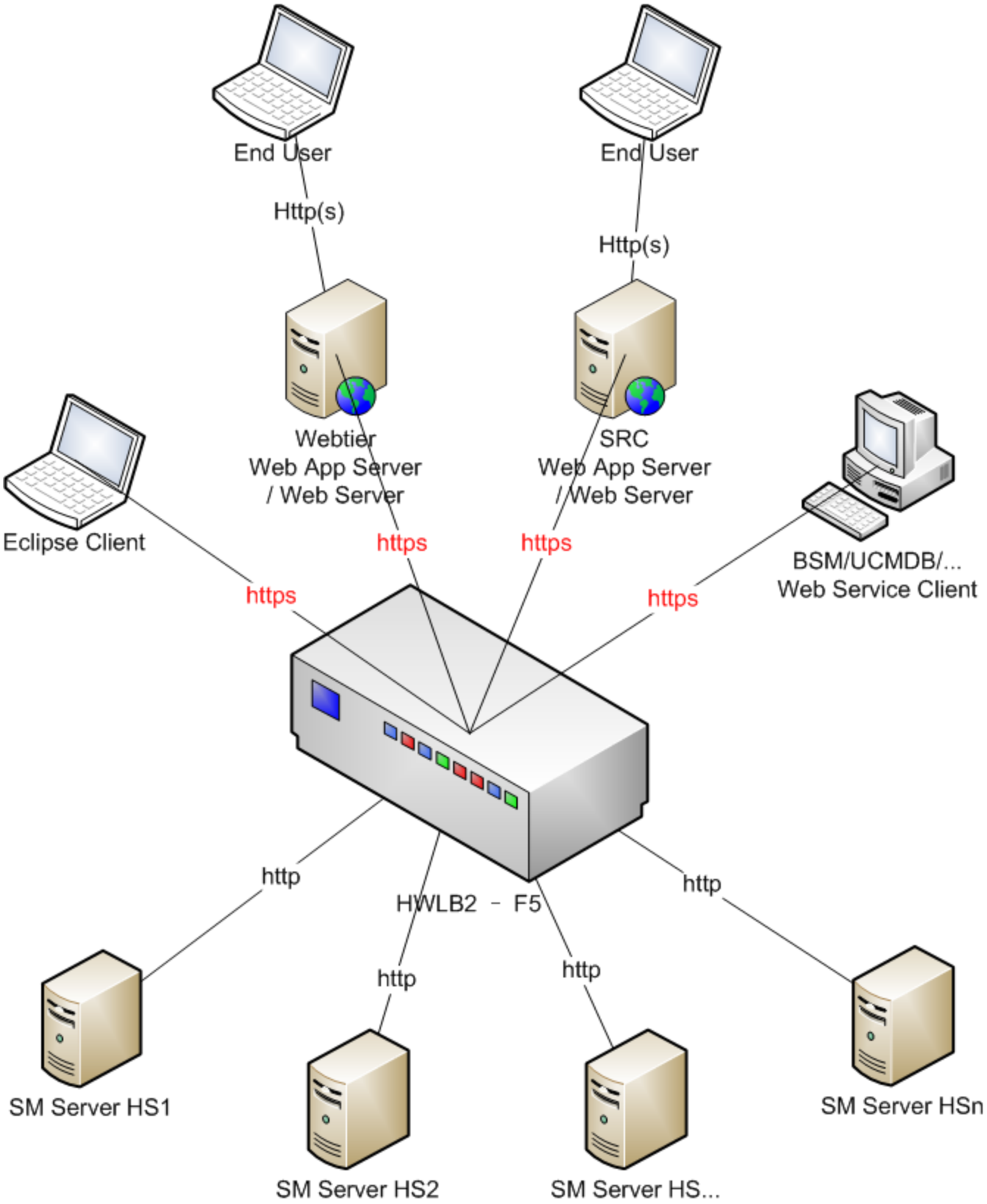
Access Service Manager through an F5 HWLB in HTTP mode

Access Service Manager on a Windows client through an F5 HWLB in HTTP mode

F5 HWLB Health Monitor

Access Service Manager through an F5 HWLB in SSL offloading mode

The following figure illustrates a deployment in which users access Service Manager through an F5 HWLB in SSL offloading mode:


End User
End User
Http(s)
Http(s)
Webtier
Web App Server
/ Web Server
SRC
Web App Server
/ Web Server
Eclipse Client
BSM/UCMDB/...
Web Service Client
https
https
https
https
HWLB2 - F5
http
http
http
http
SM Server HS1
SM Server HS2
SM Server HS...
SM Server HSn

Access Service Manager on a Windows client through an F5 HWLB in SSL Offloading mode

This section describes how to access Service Manager on a Windows client through an F5 HWLB in SSL Offloading mode.

Prerequisites

Before you begin this process, you must create the client certificates and private keys, as described in "Configure an HWLB for SSL offloading" on page 21.

Note: You do not need to configure SSL on the server.

Configure the Windows client to use SSL

To access Service Manager on a Windows client through an F5 HWLB in SSL Offloading mode, you must configure the Windows client to use SSL. To do this, follow these steps:

1. Click **Windows** > **Preference** > **HP Service Manager** > **Security**.
2. Click **Browse** to fill the **Client Keystore** file.
3. Type `clientkeystore` as the password.
4. Select the `cacerts` CA Certificates file that you created earlier, and then click **OK**.
5. Restart Windows.

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in SSL Offloading mode

Access Service Manager on a web client through an F5 HWLB in SSL Offloading mode

Access Service manager on a web service client through an F5 HWLB in SSL Offloading mode

Access Service Manager on a web client through an F5 HWLB in SSL Offloading mode

This section describes how to access Service Manager on a web client through an F5 HWLB in SSL Offloading mode.

Prerequisites

Before you begin this process, you must create the client certificates and private keys, as described in "Configure an HWLB for SSL offloading" on page 21.

Note: You do not need to configure SSL on the Service Manager server.

Modify the web.xml file

To access Service Manager on a web client through an F5 HWLB in SSL Offloading mode, configure the following parameters in the `web.xml` file, and then restart Tomcat:

- **ssl**

```
<init-param>
<param-name>ssl</param-name>
<param-value>true</param-value>
</init-param>
```

- **externalLB**

```
<init-param>
<param-name>externalLB</param-name>
<param-value>true</param-value>
</init-param>
```

- **keystore**

```
<!-- If this is a relative path, it will be relative to the web application's
deploy directory,
but still needs a leading slash -->
<init-param>
```

```
<param-name>keystore</param-name>
<param-value>/WEB-INF/<FQDN of This Host>.keystore </param-value>
</init-param>
<!-- Specify the password for the client's private keystore -->
<init-param>
<param-name>keystorePassword</param-name>
<param-value>clientkeystore</param-value>
</init-param>
```

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in SSL Offloading mode

Access Service Manager on a Windows client through an F5 HWLB in SSL Offloading mode

Access Service manager on a web service client through an F5 HWLB in SSL Offloading mode

Access Service Manager on a web service client through an F5 HWLB in SSL Offloading mode

This section describes how to access Service Manager on a web service client through an F5 HWLB in SSL Offloading mode.

Prerequisites

Before you begin this process, you must create the client certificates and private keys, as described in "Configure an HWLB for SSL offloading" on page 21.

Note: You do not need to configure SSL on the server.

Create a new project in SoapUI

To access Service Manager on a web service client through an F5 HWLB in SSL Offloading mode, follow these steps:

1. Start the SoapUI tool and click **New project**.
2. Type a name for the project in the **Project Name** field.

3. In the **Initial WSDL/WADL** field, type `http://<SM Server IP>:<SM Server Port>/SM/7/ChangeManagement.wsdl`.
4. Click to select the **Create Requests** option, and then click **OK**.
5. Right-click **RetrieveChangeList** and select **New request**.
6. Type a name for the request.
7. Click **File** > **Preference** > **SSL Settings**.
8. Navigate to the `<client FQDN>.keystore` file.
9. Type `clientkeystore` as the password, and then click **OK**.
10. Modify the request as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://schemas.hp.com/SM/7">
<soapenv:Header/>
<soapenv:Body>
<ns:RetrieveChangeListRequest attachmentInfo="?" attachmentData="?"
ignoreEmptyElements="true" count="2" start="0">
<!--1 or more repetitions:-->
<ns:keys query="planned.start=NULL">
</ns:keys>
</ns:RetrieveChangeListRequest>
</soapenv:Body>
</soapenv:Envelope>
```

11. Click **Authenticate**.
12. Type `falcon` as the user name.
13. Replace **http** in the URL with **https**. For example, set the following URL:

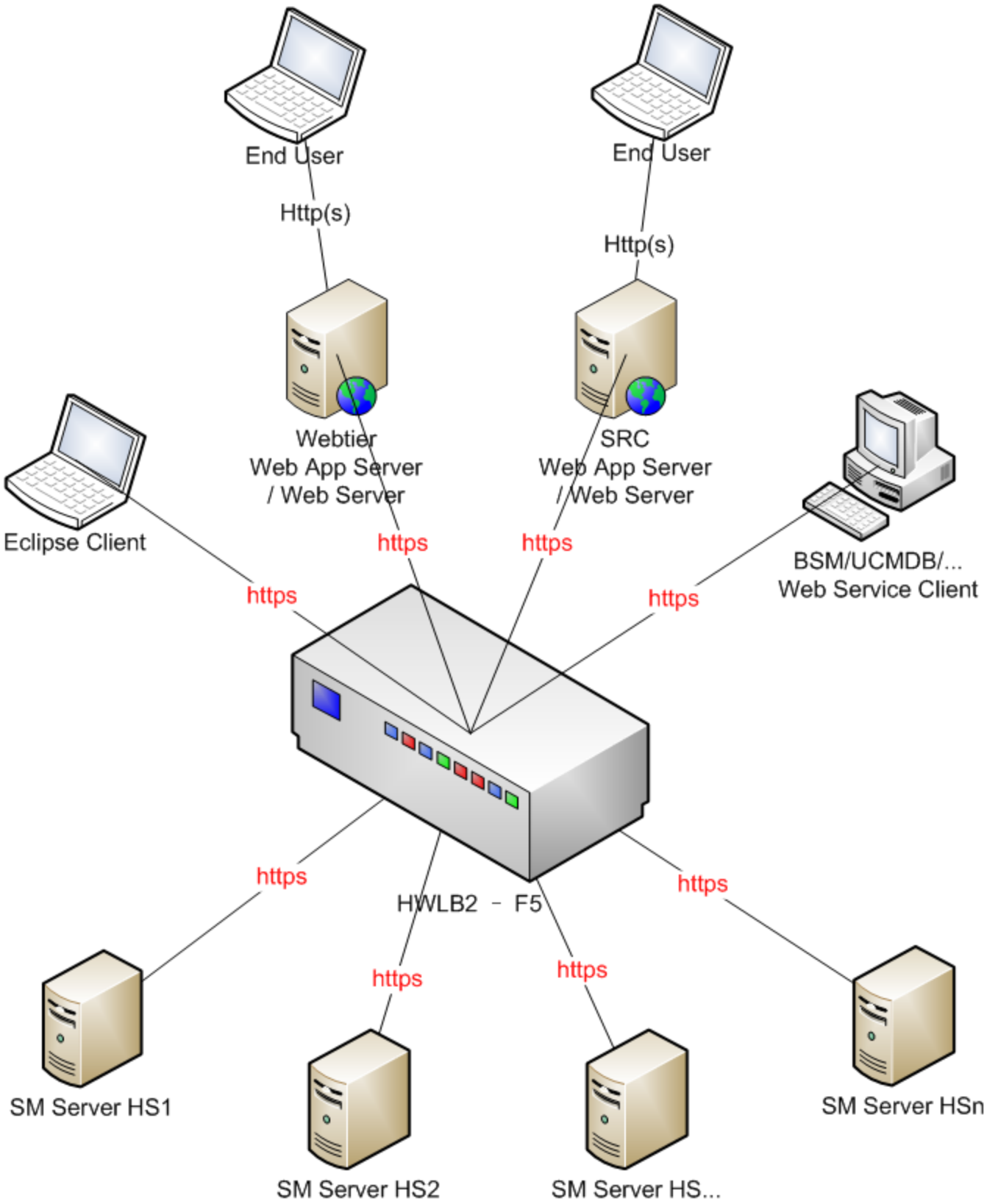
    https://<*FQDN of the HWLB virtual server*>:<*Port of HWLB virtual server*>/SM/7/ws.

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in Full SSL mode

The following figure illustrates a deployment in which users access Service Manager through an F5 HWLB in Full SSL mode:


End User
End User
Http(s)
Http(s)
Webtier
Web App Server
/ Web Server
SRC
Web App Server
/ Web Server
Eclipse Client
BSM/UCMDB/...
Web Service Client
https
https
https
https
https
https
https
https
HWLB2 - F5
SM Server HS1
SM Server HS2
SM Server HS...
SM Server HSn

Access Service Manager on a Windows client through an F5 HWLB in Full SSL mode

This section describes how to access Service Manager on a Windows client through an F5 HWLB in Full SSL mode.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- To maintain session persistence, SSL must be configured between the F5 HWLB and the Windows client. For more information about how to configure SSL, see "Configure an HWLB for SSL offloading" on page 21.
- SSL is configured between the F5 HWLB and the SM server. For more information about how to do this, see "Configure an HWLB for SSL between an F5 load balancer and a Service Manager server" on page 22 .

Modify the sm.ini file

To access Service Manager on a Windows client through an F5 HWLB in Full SSL mode, configure the following parameters in the `sm.ini` file:

```
ssl:1
ssl_reqClientAuth:2
keystoreFile:server.keystore
keystorePass:serverkeystore
ssl_trustedClientsJKS:trustedclients.keystore
ssl_trustedClientsPwd:trustedclients
truststoreFile:cacerts
truststorePass:changeit
external_lb
sslConnector:1
```

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in Full SSL mode

Access Service Manager on a web client through an F5 HWLB in Full SSL mode

This section describes how to access Service Manager on a web client through an F5 HWLB in Full SSL mode.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- To maintain session persistence, SSL must be configured between the F5 HWLB and the web client. For more information about how to configure SSL, see "Configure an HWLB for SSL offloading" on page 21.
- SSL is configured between the F5 HWLB and the SM server. For more information about how to do this, see "Configure an HWLB for SSL between an F5 load balancer and a Service Manager server" on page 22 .

Step 1: Modify the sm.ini file

Configure the following parameters in the `sm.ini` file:

```
ssl:1
ssl_reqClientAuth:2
keystoreFile:server.keystore
keystorePass:serverkeystore
ssl_trustedClientsJKS:trustedclients.keystore
ssl_trustedClientsPwd:trustedclients
truststoreFile:cacerts
truststorePass:changeit
external_lb
sslConnector:1
```

Step 2: Modify the web.xml file

On the Service Manager web tier, locate the `web.xml` file, and then configure the following parameters:

```
<init-param>
<param-name>ssl</param-name>
<param-value>true</param-value>
</init-param>
<init-param>
<param-name>cacerts</param-name>
<param-value>/WEB-INF/cacerts</param-value>
</init-param>
<init-param>
<param-name>keystore</param-name>
<param-value>/WEB-INF/<webtier FQDN>.keystore</param-value>
</init-param>
<init-param>
<param-name>keystorePassword</param-name>
<param-value>clientkeystore</param-value>
</init-param>
<init-param>
<param-name>externalLB</param-name>
<param-value>true</param-value>
</init-param>
```

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager through an F5 HWLB in Full SSL mode

Access Service Manager on a Windows client through an F5 HWLB in Full SSL mode

Access Service Manager on a web service client through an F5 HWLB in Full SSL mode

Access Service Manager on a web service client through an F5 HWLB in Full SSL mode

This section describes how to access Service Manager on a web service client through an F5 HWLB in Full SSL mode.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- To maintain session persistence, SSL must be configured between the F5 HWLB and SoapUI. For more information about how to configure SSL, see "Configure an HWLB for SSL offloading" on page 21.
- SSL is configured between the F5 HWLB and the SM server. For more information about how to do this, see"Configure an HWLB for SSL between an F5 load balancer and a Service Manager server" on page 22 .

Step 1: Modify the sm.ini file

Configure the `sm.ini` file as follows:

```
ssl:1
ssl_reqClientAuth:2
keystoreFile:server.keystore
keystorePass:serverkeystore
ssl_trustedClientsJKS:trustedclients.keystore
ssl_trustedClientsPwd:trustedclients
truststoreFile:cacerts
truststorePass:changeit
external_lb
sslConnector:1
```

Step 2: Create a new project in SoapUI

To create a new project in SoapUI, follow these steps:

1. Start the SoapUI tool and click **New project**.
2. Type a name for the project in the **Project Name** field.
3. In the **Initial WSDL/WADL** field, type `http://<HWLB virtual server IP>:13080/SM/7/ChangeManagement.wsdl`.
4. Click to select the **Create Requests** option, and then click **OK**.
5. Right-click **RetrieveChangeList** and select **New request**.
6. Type a name for the request.
7. Click **File** > **Preference** > **SSL Settings**.
8. Navigate to the `<client FQDN>.keystore` file.
9. Type `clientkeystore` as the password, and then click **OK**.
10. Modify the request as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://schemas.hp.com/SM/7">
<soapenv:Header/>
<soapenv:Body>
<ns:RetrieveChangeListRequest attachmentInfo="?" attachmentData="?"
ignoreEmptyElements="true" count="2" start="0">
<!--1 or more repetitions:-->
<ns:keys query="planned.start=NULL">
</ns:keys>
</ns:RetrieveChangeListRequest>
</soapenv:Body>
</soapenv:Envelope>
```

11. Click **Authenticate**.
12. Type `falcon` as the user name.
13. Replace **http** in the URL with **https**. For example, set the following URL:

    https://<*FQDN of the HWLB virtual server*>:<*Port of HWLB virtual server*>/SM/7/ws.

Click here to show or hide links to related topics.

Related Tasks

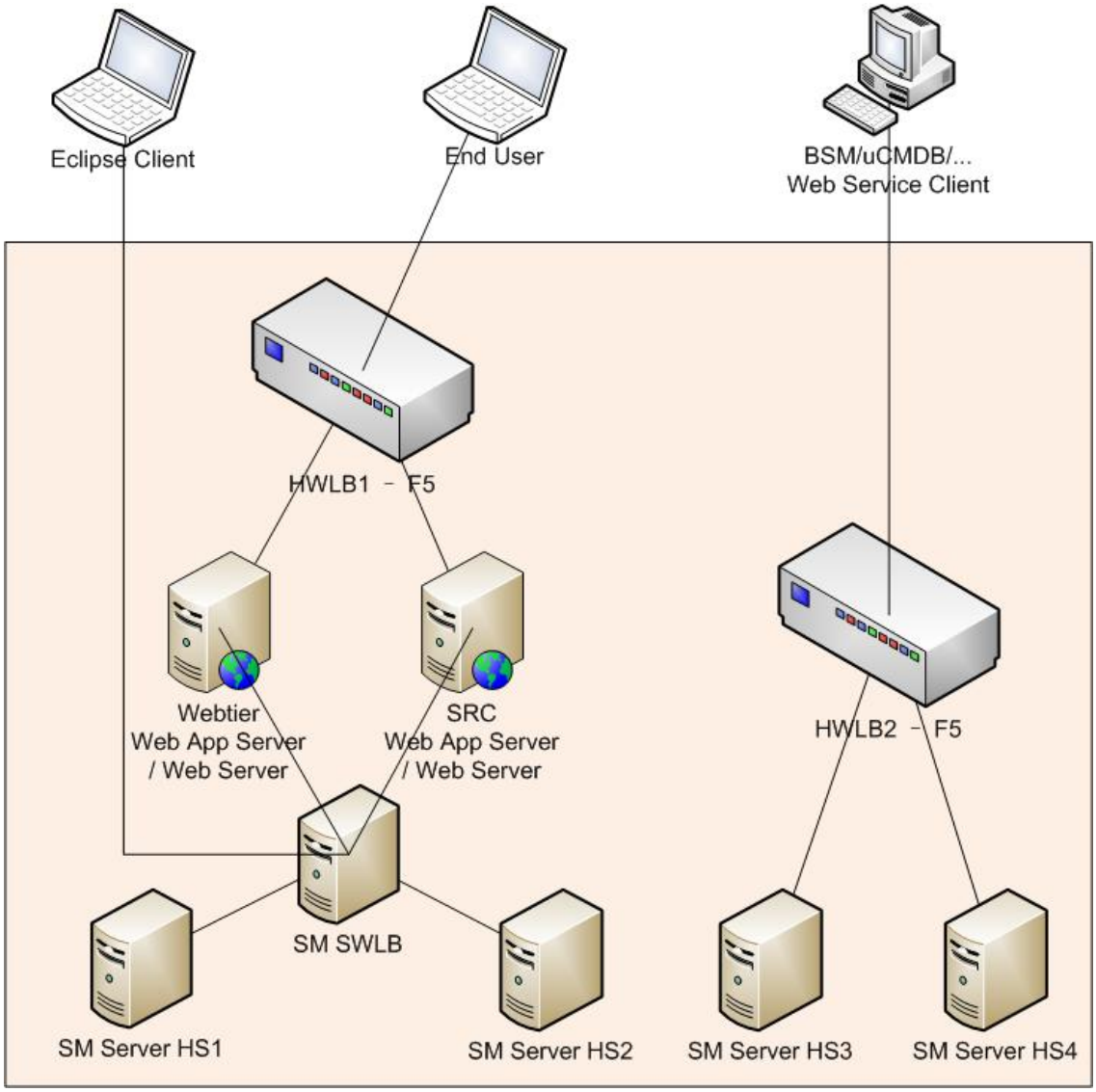
Access Service Manager through an F5 HWLB in Full SSL mode

Access Service Manager on a Windows client through an F5 HWLB in Full SSL mode

Access Service Manager on a web client through an F5 HWLB in Full SSL mode

Access Service Manager in Mixed mode

The following figure illustrates a deployment in which users access Service Manager through an F5 HWLB in Mixed mode:



Log on to Service Manager by using TSO when an HWLB is located between the browser and the web tier

This section describes how to log on Service Manager by using Trusted sign-on (TSO) when a HWLB is located between the browser and the web tier.

Prerequisites

Before you begin this process, you must access Service Manager on a web client through an F5 HWLB in Full SSL mode, as described in "Access Service Manager on a web client through an F5 HWLB in Full SSL mode" on page 39.

Configure the HWLB virtual server

To log on to a Service Manager server by using TSO, you must configure the HWLB virtual server. To do this, follow these steps:

1. Click **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Select the virtual server that you want to configure.
3. In the **HTTP Profile** field, select **TEST_HTTP**.
4. In the **OneConnect Profile** field, select **oneconnect**.
5. In the **NTLM Conn Pool** field, select **ntlm**.
6. Click **Update**.
7. Add the HWLB domain URL to your browser's trusted sites list.

You can now log on to the Service Manager web tier through the HWLB. For example, type the following URL into the address bar of your browser:

`http://<FQDN of HWLB virtual server>/sm932/index.do`

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager in Mixed mode

Log on to a Service Manager server by using LW-SSO when a HWLB is located between the browser and the webtier

Log on to Service Manager by using LW-SSO when an HWLB is located between the browser and the web tier

This section describes how to log on Service Manager by using Lightweight Single Sign-On (LW-SSO) when an HWLB is located between the browser and the web tier.

Prerequisites

Before you begin this process, verify that the following conditions are true:

- The HWLB is configured to accept client requests, as described in"Configure an HWLB to accept client requests in HTTP mode" on page 17.
- A pool contains the members that point to the web application server (for example, Tomcat).
- The server and the web tier share the same initString setting.
- LW-SSO works without the HWLB.

> **Note:** If the domain name of the HWLB is not the same as the domain name of the web tier, you must set the value of the `domain` parameter in the `lwssofmconf.xml` file on the web tier to the domain name of the HWLB.

Enable LW-SSO

1. Enable LW-SSO on the Service Managerserver. For more information about how to do this, see "Configure LW-SSO in the server" on page 1.
2. Enable LW-SSO on the web tier. For more information about how to do this, see "Configure LW-SSO in the Web tier" on page 1.

Click here to show or hide links to related topics.

Related Tasks

Access Service Manager in Mixed mode

Log on to a Service Manager server by using TSO when a HWLB is located between the browser and the webtier

Horizontal scaling implementation

In a horizontal scaling implementation you maximize the number of client connections supported across multiple hosts.

The key features of this implementation are:

- The HP Service Manager system consists of multiple Service Manager servlet container processes running on separate physical hosts
- The implementation allows administrators to specify a system connection limit
- The Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threads per process value (For example, 6 servlet container processes supporting 50 threads per process can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses
- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove Service Manager instances from a virtual group

A horizontal scaling implementation is typically used in large 24 by 7 environments where system scalability and resilience is a concern. A horizontal scaling implementation can support as many client connections as the sum of the individual Service Manager instances can support (for example, if each instance can support 50 client connections and there are 6 instances then the total system can support 300 client connections). A horizontal scaling implementation has improved resilience features such as the ability to stop and start the load balancer process without causing a total system outage and the ability to add a new servlet container process to a virtual group while the system is running.

Each host in a horizontal scaling implementation can also support a vertical scaling implementation with the addition of the appropriate servlet implementation parameters. This allows each Service Manager host to support multiple servlet container processes and to make the best use of the available system resources.

Note: There can be only **one** kmupdate process running at any time regardless of the number of hosts. Starting more than one kmupdate process causes unpredictable behavior on the search engine server.

Horizontal scaling implementation diagram

The *Horizontal scaling implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for multiple servers running multiple servlet containers implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Horizontal scaling implementation diagram* is available from the help.

Example: Setting up a horizontal scaling implementation

The following example describes how to set up a horizontal scaling implementation that accomplishes the following:

- Maximizes the number of client connections supported across multiple hosts
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses

You can use this example to configure the implementation depicted in the *Horizontal scaling implementation diagram*.

1. Install Service Manager on the first host in the virtual group. For example: myserver1.
   The host must be on the same subnet as the other hosts in the virtual group.
2. Log in to the Webware software licensing center and obtain a license file for this host.

   Note: After you obtain a license file for this host, it becomes the primary host of the horizontal scaling implementation. You must start this host first when starting a horizontal scaling implementation for the first time.
3. Copy the license file you obtain onto a network share accessible to the other servers that will make up the horizontal scaling implementation.
4. Log in to the operating system of the primary host and change directories to the server's AutoPass folder. For example:

```
C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
```

5. Paste the license file (LicFile.txt) into the AutoPass folder.
6. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\Service Manager\server\RUN
   ```

7. Open the Service Manager configuration file (sm.cfg) in a text editor.
8. Edit the file so that only the following lines appear:

   ```
   sm -loadBalancer -httpPort:13080
   sm -que:ir
   sm -httpPort:13081 -httpsPort:13082
   sm -httpPort:13083 -httpsPort:13084
   sm system.start
   ```

   The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A horizontal scaling implementation only needs one load balancer.

   The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.

   The que:ir parameter starts the processing of scheduled IR records that the server generates running in asynchronous IR mode.

   The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

   The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

   The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The system.start parameter specifies that the host system should start all the background processes in the system.start script file. This host will be the only host in the virtual group that runs the background processes. The configuration file should already list this line.

9. Save the Service Manager configuration file.
10. Open the Service Manager initialization file (sm.ini) in a text editor.
11. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

12. Add the following lines:

```
grouplicenseip:10.0.0.135
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupsubnetaddress:255.255.255.0
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The communications port can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

13. Save the Service Manager initialization file.
14. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

```
127.0.0.1 localhost
127.0.0.2 myserver2.mydomain.com myserver2 loghost
127.0.0.3 myserver3.mydomain.com myserver3 loghost
```

Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

15. Start the Service Manager server.

16. Install Service Manager on the next host in the virtual group. For example: myserver2.
    The host must be on the same subnet as the other hosts in the virtual group.

17. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

    ```
    C:\Program Files\HP\Service Manager\server\RUN
    ```

18. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.

19. Edit the file so that only the following lines appear:

    ```
    sm -httpPort:13081 -httpsPort:13082
    sm -httpPort:13083 -httpsPort:13084
    sm -sync
    ```

    Note: You do not need the system.start line on this host as only one host in the virtual group needs to run the background processes.

    The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

    The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

    The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

    The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The sync parameter specifies that the host system should start the sync background process to identify and remove unused processes.

20. Save the Service Manager configuration file.

21. Open the Service Manager initialization file (sm.ini) in a text editor.

22. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

23. Add the following lines:

```
grouplicenseip:10.0.0.135
groupname:mycluster1
groupmcastaddress:224.0.1.255
groupsubnetaddress:255.255.255.0
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The TCP/IP address can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

24. Save the Service Manager initialization file.

25. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

    ```
    127.0.0.1 localhost
    127.0.0.2 myserver2.mydomain.com myserver2 loghost
    127.0.0.3 myserver3.mydomain.com myserver3 loghost
    ```

    Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

26. Log in to the operating system of the Service Manager server host and copy the primary host's license file (LicFile.txt) from the network share.

27. Change directories to this server's AutoPass folder. For example:

    ```
    C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
    ```

28. Paste the primary host's license file into this server's AutoPass folder.
29. Start the Service Manager server.

    Note: The primary host must be running in order for this host to validate the license file and start up.

30. Install Service Manager 9.41 on the next host in the virtual group. For example: myserver3.
    The host must be on the same subnet as the other hosts in the virtual group.
31. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

    ```
    C:\Program Files\HP\Service Manager 9.41\server\RUN
    ```

32. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
33. Edit the file so that only the following lines appear:

    ```
    sm -httpPort:13081 -httpsPort:13082
    sm -httpPort:13083 -httpsPort:13084
    sm -sync
    ```

    Note: You do not need the system.start line on this host as only one host in the virtual group needs to run the background processes.

    The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

    The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

    The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

    The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The sync parameter specifies that the host system should start the sync background process to identify and remove unused processes.

34. Save the Service Manager configuration file.

35. Open the Service Manager initialization file (sm.ini) in a text editor.

36. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

37. Add the following lines:

```
grouplicenseip:10.0.0.135
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupsubnetaddress:255.255.255.0
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The TCP/IP address can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

38. Save the Service Manager initialization file.

39. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

    ```
    127.0.0.1 localhost
    127.0.0.2 myserver2.mydomain.com myserver2 loghost
    127.0.0.3 myserver3.mydomain.com myserver3 loghost
    ```

    Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

40. Log in to the operating system of the Service Manager server host and copy the primary host's license file (LicFile.txt) from the network share.

41. Change directories to this server's AutoPass folder. For example:

    ```
    C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
    ```

42. Paste the primary host's license file into this server's AutoPass folder.
43. Start the Service Manager server.

    Note: The primary host must be running in order for this host to validate the license file and start up.
44. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

    ```
    C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
    ```
45. Open the Web configuration file (web.xml) in a text editor.
46. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
```

```
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

47. Save the Web configuration file.
48. Start the Service Manager Web tier.
49. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080

50. Connect to the Service Manager host.

Configuring a horizontal scaling environment

This configuration is intended for customers who:

- Want to maximize the number of client connections supported across multiple hosts
- Have multiple hosts available to manage all concurrent client connections

Terminology

- **Group**: A group of Service Manager server processes running on one or more hosts and connecting to one database to serve Service Manager server processes within a group.
- **Node**: A Service Manager server process within a group.
- **Primary host**: The host in the Service Manager horizontal scaling group whose IP is bound to the Service Manager AutoPass license.
- **Secondary host**: The host in the Service Manager horizontal scaling group whose IP is not bound to the Service Manager AutoPass license. The Service Manager processes on this host depend on

running Service Manager processes on the primary host to validate the Service Manager AutoPass license.

- **Service Manager servlet process**: A Service Manager server process that embeds the Tomcat application server and serves Service Manager Windows clients and Web clients, as well as Web Services requests. It is also generally referred to as a Service Manager process.
- **Service Manager background process**: Background processes that wake up periodically to execute a particular RAD application, or Service Manager runtime environment routine. For example: `sm -que:ir`.
- **Service Manager transient process**: A Service Manager server process that executes a RAD application or Service Manager runtime environment routine only once, not periodically. For example: `sm -reportlbstatus`.

Number of Service Manager hosts required

This configuration can be set up on one or more hosts.

Commands required in sm.cfg on primary host

You must set the following configuration commands.

```
sm -loadBalancer -httpPort:<value>
sm -que:ir
sm -httpPort:<value> -httpsPort:<value>
sm system.start
```

Commands required in sm.cfg on secondary host

```
sm -httpPort:<value> -httpsPort:<value>
sm -sync
```

- sm -loadBalancer -httpPort:<value> – Is a special Service Manager process that redirects client connections requests to other available Service Manager processes.
- sm -httpPort:<value> -httpsPort:<value> – Specifies the HTTP or HTTPS communications port for one Service Manager process. Each Service Manager process by default can host 50 users. You can start as many Service Manager processes as you want, as long as your server has enough physical memory and resources to start the processes. You can also change the number of users hosted on one Service Manager process by setting the "threadsperprocess" parameter.

- sm -que:ir – Starts the processing of scheduled IR records that the server generates running in asynchronous IR mode. Since asynchronous IR mode is enforced in a horizontal scaling environment, there has to be only one "sm -que:ir" process for the horizontal scaling group.
- sm system.start – Starts one Service Manager process that contains all background processes.
- sm -sync – Identifies defunct Service Manager processes on the host and frees the shared resources that were allocated for it. The "sm -sync" process needs to start only one per host. Since the "sm system.start" process includes the "sm -sync" background process, the "sm -sync" command should not be included in the `sm.cfg` file that has the "sm system.start" command.

Parameters required in sm.ini

You must set the following initialization parameters.

```
system:<value>
grouplicenseip:<value>
groupname:<value>
groupmcastaddress:<value>
groupport:<value>
```

- system – Defines a unique numerical ID for the system. The value of this parameter must be identical in the `sm.ini` files for each host in the horizontal scaling group.
- grouplicenseip – The value should be the primary host IP.
- groupname – Creates a group name as the horizontal scaling group identifier that the Service Manager processes use to identify the group. The value of this parameter is only alphanumeric characters.
- groupmcastaddress – Defines the TCP/IP multicast address that all Service Manager processes within the horizontal scaling group use to communicate with each other.
- groupport – Defines the communications port that all Service Manager processes use to communicate with all other Service Manager processes within the horizontal scaling group.
- [RDBMS Settings] – Define the RDBMS connection and authorization parameters.

Optional parameters in sm.ini

The following initialization parameters are optional.

```
threadsperprocess:<value>
preferredFQHN:<value>
groupbindaddress:<value>
```

- threadsperprocess – Defines the maximum number of concurrent user sessions per Service Manager process. Use a value that maximizes the system resources of your Service Manager host. The recommended maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.
- preferredFQHN – Specifies the fully qualified host name you want Service Manager clients to use when communicating with the server. Service Manager loadBalancer redirects client requests to the target host with the target host's "preferredFQHN." You only need to set this parameter if your Service Manager host is identified by multiple names in the network.
- groupbindaddress – Defines the TCP/IP address of the network adapter you want Service Manager processes to use to communicate with other processes in a horizontal scaling group. If your Service Manager hosts contain multiple network adapters, you must specify the IP address of the network adapter you want the horizontal scaling group to use with the "groupbindaddress" parameter.

Operating system requirements

All hosts in the horizontally-scaled environment must run on the same operating system version and run at the same patch level. Minimum patch level for each operating system is specified in the support matrix. See HP Support Matrices" on the Software Support Online site.

Caution: Running Service Manager in a horizontally-scaled environment and using different operating systems can corrupt your data. You can also run into problems when upgrading to the latest patches. Most of the time, Service Manager patches are released on all operating systems. However, in some cases, patches may be delivered to a particular operating system. When mixing operating systems, applying patches will become more complex as you need to test on multiple platforms.

Network requirements

All hosts in the horizontal scaling group must run from the same subnet on the network in order for the Service Manager processes to communicate with one another.

Memory requirements

HP recommends around 1.5 GB RAM per Service Manager process. This is based on the expected user load. For example, if you have 10 processes each running with a 50-user load, the minimum required RAM would be 1.5 GB per Service Manager process, 10x1.5 GB RAM. For information on sizing, refer to the *Service Manager 7 Reference Configurations* sizing guide in the HP Software online support knowledge documents at the following URL: www.hp.com/go/hpsoftwaresupport.

Licensing requirements

Obtain an AutoPass license for one host in your horizontal scaling group. On Windows platforms, AutoPass installs as part of the server installation. On Unix platforms, you must install it manually

before you can run Service Manager. For more information, see the *HP Service Manager Installation Guide* in the related topics. This host becomes the primary host of the horizontal scaling group and AutoPass includes this host's IP address in the `LicFile.txt` file. You must start the primary host first when starting the horizontal scaling group. Copy the `LicFile.txt` file from the primary host's AutoPass directory to the AutoPass directory of each secondary host in the horizontal scaling group. Each secondary host must have a copy of the primary host's `LicFile.txt` file in order to start.

The summary steps are as follows:

- List all hosts in your horizontal scaling environment to determine the primary server and all secondary servers.
- Get the license from the HP Web site.
- Copy the primary server's IP address to get the license and save it as the primary server's `LicFile.txt` file.
- Copy the primary server's `LicFile.txt` file on all secondary servers. The default directory for this file is as follows:

  `<Service Manager server installation path>/RUN/LicFile.txt`

- Add the following parameter in the `sm.ini` file: `grouplicenseip:Primary_Servers_IpAddress`

To start the horizontal scaling group

1. Start the Service Manager server on the primary server.

   Note: The primary server must be started prior to starting any of the secondary servers.

2. Start the Service Manager server on the secondary servers, in no particular order.

To establish the horizontal scaling group, the first node of the group should be started on the primary host. When the group is established on the primary host, Service Manager server processes on secondary hosts can start to join the group. Once the whole group is established, the primary host can be brought down for maintenance, and then rejoin the rest of the running group when it is restarted. As long as there is a member Service Manager server process running in the group, Service Manager server processes on another secondary host with the same "grouplicenseip:<primary host IP>" can join the group. The primary host is not required to be running. Another secondary host can be started, even if the primary host is down for any reason. However, when the group is down (for example, there are no nodes running in the group), then the group has to be reestablished from the primary host first.

Other requirements

Install the applicable database client software on each host of the horizontal scaling group so that they can access the Service Manager RDBMS. See your RDBMS vendor documentation for instructions.

Configuring SSL and LW-SSO in a horizontal scaling environment

In a horizontally scaled environment, you use the Service Manager software load balancer in front of a virtual group to redirect client requests among the server nodes in the group. The load balancer server is called the primary server, while the other servers in the group are called secondary servers.

Important Requirements

- On a horizontally scaled system, it is very important that all certificates are created from the same machine. Ensure to not copy any files to their target directories until all certificates for all server machines (the primary server and secondary server machines) in the horizontally scaled environment are created.
- For each server host (primary or secondary), you need a unique server certificate.
- For each Windows client machine you need a unique client certificate.
- For each Service Manager Web application server host you need a unique client certificate.

Best Practice recommendation:

When generating each Windows or web client certificate, enter the FQDN name of the client machine in front of the keystore, certificate request and certificate names to make them unique and easy to distinguish.

If a Windows client and the Web application server are on the same physical machine, it is possible to use the same cacerts and clientcerts files for both, rather than creating two sets of nearly identical keystores. In such a case, copy the files created for the Windows or Web client – whichever was created first– into either the <Service Manager Client>/plugins/com.hp.ov.sm.client.common_x.xx directory or the Service Manager/WEB-INF folder of the Web application server.

To configure SSL in a horizontal scaling implementation:

1. Create a private key and public certificate for your private certificate authority. For details, see "Example: Generating a server certificate with OpenSSL" on page 1.
2. Update the Java cacerts file by importing your private certificate authority's certificate into it. The

updated cacerts file will contain the certificate and private key of the certificate authority that signs each server/client certificate. For details, see "Example: Generating a server certificate with OpenSSL" on page 1.

3. Create a separate signed server keystore for each server machine. For details, see "Example: Generating a server certificate with OpenSSL" on page 1.

   Note: In a later step, you will add each server keystore and the updated Java cacerts file to each server's sm.ini file as the following parameters: **keystoreFile**, and **truststoreFile**.

4. Create a separate signed client certificate for each Service Manager client (web application server or Windows client) machine and import each client certificate into a trusted clients keystore. For details, see "Example: Generating a client certificate with OpenSSL" on page 1.

   Note: In a later step, you will add the trusted clients keystore as the following parameter in each server's sm.ini file: **ssl_trustedClientsJKS**.

5. Copy the following keystore files to each server's RUN folder:
   - The server's server keystore file (unique)
   - The trusted clients keystore file (same for all servers)
   - The updated Java cacerts file (same for all servers and clients)
6. Copy the following keystore files to each web tier's WEB-INF folder:
   - The updated Java cacerts file
   - <clientcerts>.keystore – This keystore contains the signed certificate of your Service Manager web tier client.
7. Copy the following keystore files to each Windows client's <Windows client installation path>\plugins\com.hp.ov.sm.client.common_x.xx folder.
   - The updated Java cacerts file
   - <clientcerts>.keystore – This keystore contains the signed certificate of each Windows client.
8. Add the following SSL configuration lines to the sm.ini file for each of the primary server and secondary servers, and restart each server.

Note: For all servers, the truststoreFile and ssl_trustedClientsJKS settings should be the same; however, each server should use its own server keystore (keystoreFile:<servercert.keystore>).

```
ssl:1
ssl_reqClientAuth:2
sslConnector:1
ssl_trustedClientsJKS:<trustedclients.keystore>
ssl_trustedClientsPwd:<ClientKeyPassword>
truststoreFile:cacerts
truststorePass:changeit
keystoreFile:<servercert.keystore>
keystorePass:changeit
```

9. Configure each Service Manager web client to validate each server's signed certificate and present the signed client certificate.
   a. Stop the web application server running the web tier, open the web configuration file (web.xml) in a text editor.
   b. Configure the following parameters as shown below:

```
<init-param>
    <param-name>cacerts</param-name>
    <param-value>/WEB-INF/cacerts</param-value>
</init-param>
<init-param>
    <param-name>keystore</param-name>
    <param-value>/WEB-INF/<clientcerts>.keystore</param-value>
</init-param>
<init-param>
    <param-name>keystorePassword</param-name>
    <param-value><<clientcerts>.keystore password></param-value>
</init-param>
```

   c. Set the ssl parameter to true.

d. Set the serverHost and serverPort parameters to the fully-qualified domain name and port number of the primary server. For example: myprimaryserver.mydomain.com and 13080.

e. Restart each web application server.

10. Configure each Windows client to validate each server's signed certificate and present the signed client certificate.

a. Click **Window** > **Preferences** > **Service Manager** > **Security**.

b. Set CA Certificates File to the cacerts keystore you copied to the <Windows client installation path>\plugins\com.hp.ov.sm.client.common_x.xx folder.

c. Set Keystore File to the keystore containing your Windows client's signed certificate, for example <clientcerts>.keystore. You created this keystore when you created the Windows client certificate request.

d. Set Keystore password to the password required to access the Windows client keystore. For example, ClientKeyPassword. You created this keystore password when you created the Windows client certificate request.

e. Update your Windows client connections by selecting **Use SSL Encryption** on their Advanced tab.

To set up LW-SSO in a horizontal scaling implementation:

1. Enable LW-SSO on each server node. All server nodes should share the same initString setting. For details, see Configure LW-SSO in the server.

   Note: Normally, all server nodes are in the same domain, so you can configure LW-SSO in one server node and then copy the server's LW-SSO configuration file (lwssofmconf.xml) to the rest of the nodes.

2. If needed, enable LW-SSO in each web tier. For details, see Configure LW-SSO in the Web tier.

   Note: The server nodes and web tiers should share the same initString setting.

Single servlet implementation

In a single servlet implementation you manage all client connections with one multithreaded process.

The key features of this implementation are:

- The HP Service Manager system consists of one HP Service Manager instance running on one physical host
- One dedicated servlet container process manages all client connections
- The implementation allows administrators to specify the communication ports the system uses
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of threadsperprocess

A single servlet implementation is a typically used for development environments or small production environments because it is easy to setup and manage.

You can convert a single servlet implementation into any of the other servlet implementations with the addition of a load balancer process and one or more additional servlet container instances.

Single servlet implementation diagram

The *Single servlet implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running a single servlet container implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Single servlet implementation diagram* is available from the help.

Example: Setting up a single servlet implementation

The following example describes how to set up a single servlet implementation that accomplishes the following:

- Manages all client connections with one multithreaded process
- Supports up to 50 concurrent client connections

- Allows an administrator to specify the communications ports the HP Service Manager implementation uses
- Provides a simple configuration to test servlet features

You can use this example to configure the implementation depicted in the *Single servlet implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\ Service Manager\server\RUN
   ```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following line:

   ```
   sm -httpPort:13081 -httpsPort:13082
   ```

   The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.
   The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.
5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Add the RDBMS connection settings. For example:

   ```
   [oracle10]
   sqldb:ora102
   sqllogin:sm7user/password
   sqllibrary:SQORACLE.OCI10.DLL
   sqldictionary:oracle10
   ```

8. Add the following lines:

   ```
   threadsperprocess:50
   sessiontimeout:3
   ```

   The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended

system memory available for servlet container processes.
The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

9. Save the Service Manager initialization file.
10. Start the Service Manager server.
11. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

    ```
    C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
    ```

12. Open the Web configuration file (web.xml) in a text editor.
13. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
```

```
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

14. Save the Web configuration file.
15. Start the Service Manager Web tier.
16. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13081

17. Connect to the Service Manager host.

Requirements for a single servlet implementation

This configuration is intended for customers who:

- Want to manage all client connections with one multithreaded process
- Only expect up to 50 concurrent client connections
- Want to specify the communications ports the HP Service Manager implementation uses
- Want a simple configuration to test servlet features

Number of Service Manager hosts required

This implementation requires the following number of hosts.

1

Parameters required in sm.cfg

You must set the following configuration parameters.

```
sm -httpPort:<value> - httpsPort:<value>
```

- httpPort – identify the communications port that a servlet container process uses to communicate with clients using HTTP. The servlet container communications port must be unique on the host on which you start the servlet container.
- httpsPort – identify the communications port that a servlet container process uses to communicate with clients using HTTPS. The servlet container communications port must be unique on the host on which you start the servlet container.

Parameters required in sm.ini

You must set the following initialization parameters.

- sessiontimeout – define the number of minutes a client connection can remain unresponsive before the server closes the connection.
- threadsperprocess – identify the total number of threads the servlet container process supports. Use a value of threads that maximizes the system resources of your Service Manager host.
  The recommend maximum value for the parameter threads per process is 60. Usually the value of this parameter should be below 50.

Parameters required in web.xml

You must set the following Web parameters.

- serverHost – identify the host name of the Service Manager host
- serverPort – identify the communications port on which the Service Manager host listens for client connections requests

Windows client preferences required

You must set the following preferences from the **Connection** menu.

- Server host name – identify the host name of the Service Manager host
- Server port number – identify the communications port on which the Service Manager host listens for client connections requests

Vertical scaling implementation

In a vertical scaling implementation you maximize the number of client connections supported on a single host.

The key features of this implementation are:

- The HP Service Manager system consists of several servlet container processes running on one physical host
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threadsperprocess value (For example, 6 servlet container processes supporting 50 threadsperprocess can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses
- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove HP Service Manager instances from a virtual group

A vertical scaling implementation is typically used in small to medium environments where hardware system resources are limited. A vertical scaling implementation can support as many client connections as the HP Service Manager host has available system resources.

You can convert a vertical scaling implementation into a horizontal scaling implementations with the addition of virtual grouping parameters and one or more additional servlet container instances installed on separate physical hosts.

Note: There can be only **one** kmupdate process running at any time regardless of the number of hosts. Starting more than one kmupdate process causes unpredictable behavior on the search engine server.

Vertical scaling implementation diagram

The *Vertical scaling implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running multiple servlet containers implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Vertical scaling implementation diagram* is available from the help.

Example: Setting up a vertical scaling implementation

The following example describes how to set up a vertical scaling implementation that accomplishes the following:

- Maximizes the number of client connections supported on a single host
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses

You can use this example to configure the implementation depicted in the *Vertical scaling implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the HP Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\ Service Manager\server\RUN
   ```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following lines:

   ```
   sm -loadBalancer -httpPort:13080
   sm -httpPort:13081 -httpsPort:13082
   sm -httpPort:13083 -httpsPort:13084
   ```

   The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A vertical scaling implementation only needs one load balancer.

   The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.

   The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

   The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

   The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on

which you start the servlet container.
The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Verify that the auth parameter has a valid authorization code.
8. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

9. Add the following lines:

```
threadsperprocess:50
sessiontimeout:3
```

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.
The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

10. Save the Service Manager initialization file.
11. Start the Service Manager server.
12. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

```
C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
```

13. Open the Web configuration file (web.xml) in a text editor.

14. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

15. Save the Web configuration file.

16. Start the Service Manager Web tier.

17. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080

18. Connect to the Service Manager host.

Configuring a vertical scaling environment

This configuration is intended for customers who:

- Want to maximize the number of client connections supported on a single host
- Have a host with enough system resources to manage all concurrent client connections

Terminology

- **Group**: A group of Service Manager server processes running on one or more hosts and connecting to one database to serve Service Manager server processes within a group.
- **Node**: A Service Manager server process within a group.
- **Service Manager servlet process**: A Service Manager server process that embeds the Tomcat application server and serves Service Manager Windows clients and Web clients, as well as Web Services requests. It is also generally referred to as a Service Manager process.
- **Service Manager background process**: Background processes that wake up periodically to execute a particular RAD application, or Service Manager runtime environment routine. For example: `sm -que:ir`.
- **Service Manager transient process**: A Service Manager server process that executes a RAD application or Service Manager runtime environment routine only once, not periodically. For example: `sm -reportlbstatus`.

Commands required in sm.cfg

You must set the following configuration commands.

```
sm -loadBalancer -httpPort:<value>
sm -httpPort:<value> -httpsPort:<value>
sm system.start
```

- sm -loadBalancer -httpPort:<value> – Is a special Service Manager servlet process that redirects client connections requests to other available Service Manager processes.
- sm -httpPort:<value> -httpsPort:<value> – Specifies the HTTP or HTTPS communications port for one Service Manager servlet process. Each Service Manager process by default can host 50 users. You can start as many Service Manager servlet processes as you want, as long as your server has enough physical memory and resources to start the processes. You can also change the number of users hosted on one Service Manager servlet process by setting the "threadsperprocess" parameter.

Optional parameters in sm.ini

The following initialization parameters are optional.

```
threadsperprocess:<value>
preferredFQHN:<value>
groupbindaddress:<value>
```

- threadsperprocess – Defines the maximum number of concurrent user sessions per Service Manager servlet process. Use a value that maximizes the system resources of your Service Manager host. The recommended maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.
- preferredFQHN – Specifies the fully qualified host name you want Service Manager clients to use when communicating with the server. Service Manager loadBalancer redirects client requests to the target host with the target host's "preferredFQHN." You only need to set this parameter if your Service Manager host is identified by multiple names in the network.
- groupbindaddress – Defines the TCP/IP address of the network adapter you want Service Manager processes to use to communicate with other processes in the group.

Memory requirements

HP recommends around 1.5 GB RAM per Service Manager process. This is based on the expected user load. For example, if you have 10 processes each running with a 50-user load, the minimum required RAM would be 1.5 GB per Service Manager process, 10x1.5 GB RAM. For information on sizing, refer to the *Service Manager 7 Reference Configurations* sizing guide in the HP Software online support knowledge documents at the following URL: www.hp.com/go/hpsoftwaresupport.

Licensing requirements

Obtain an AutoPass license for your vertical scaling environment. On Windows platforms, AutoPass installs as part of the server installation. On Unix platforms, you must install it manually before you can

run Service Manager. For more information, see the *HP Service Manager Installation Guide* in the related topics. AutoPass includes this host's IP address in the `LicFile.txt` file.

Do the following:

- Get the license from the HP Web site.
- Copy the primary server's IP address to get the license and save it as the primary server's `LicFile.txt` file. The default directory for this file is as follows:
  - **On Windows**: C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data\LicFile.txt
  - **On Unix**: /var/opt/OV/HPOvLIC/LicFile.txt

Vertical scaling and required SSL implementation

In a vertical scaling and required SSL implementation you maximize the number of client connections supported on a single host and provided SSL-encrypted security between client and server communications.

The key features of this implementation are:

- The HP Service Manager system consists of several servlet container processes running on one physical host
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threadsperprocess value (For example, 6 servlet container processes supporting 50 threadsperprocess can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses
- The implementation requires SSL-encrypted communications between HP Service Manager clients and servers
- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove HP Service Manager instances from a virtual group

A vertical scaling implementation is typically used in small to medium environments where hardware system resources are limited. A vertical scaling implementation can support as many client connections as the HP Service Manager host has available system resources.

You can convert a vertical scaling implementation into a horizontal scaling implementations with the addition of virtual grouping parameters and one or more additional servlet container instances installed on separate physical hosts.

Vertical scaling and required SSL implementation diagram

The *Vertical scaling and required SSL implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running multiple servlet containers in an required SSL implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Vertical scaling and required SSL implementation diagram* is available from the help.

Example: Setting up a vertical scaling and required SSL implementation

The following example describes how to set up a vertical scaling and required SSL implementation that accomplishes the following:

- Maximizes the number of client connections supported on a single host
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses
- Require SSL encryption for all connections
- Protects against complex SSL-related attacks
- Authenticates that the Service Manager server is a valid host

You can use this example to configure the implementation depicted in the *Vertical scaling and required SSL implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

```
C:\Program Files\HP\Service Manager x.xx\Server\RUN
```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following lines:

```
sm -loadBalancer -httpPort:13080
sm -httpPort:13081 -httpsPort:13082
sm -httpPort:13083 -httpsPort:13084
```

   - The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A vertical scaling implementation only needs one load balancer.
   - The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.
   - The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.
   - The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.
   - The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.
   - The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
```

```
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

8. Add the following lines:

```
threadsperprocess:50
sessiontimeout:3
truststoreFile:cacert.keystore
truststorePass:<cacert password>
keystoreFile:scserver.keystore
keystorePass:<server certificate password>
ssl:1
```

   - The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

   - The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

   - The truststoreFile:cacert.keystore parameter defines the file name and path to the keystore containing a list of trusted CA certificates. This value assumes you are using the default trust store file provided in the Service Manager server's RUN folder. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to find the trust store file.

   - The truststorePass:<cacert password> parameter specifies the password to the trust store file. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to read the password value.

   - The keystoreFile:scserver.keystore parameter defines the file name and path to the keystore containing the server's certificate file and private key. This value assumes you are using the default trust store file provided in the Service Manager server's RUN folder. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to find the keystore file.

   - The keystorePass:<server certificate password> parameter specifies the password to the keystore file. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to read the password value.

   - The ssl:1 parameter requires the Service Manager server to use a signed server certificate for SSL-encryption of all client-server communications. Each client connection validates the

server's certificate against the signing certificate authority. You must also use the keystoreFile and keystorePass parameters to define the location of the server certificate and private key.

9. Save the Service Manager initialization file.
10. Start the Service Manager server.
11. Log in to the operating system of the HP Service Manager Web tier host and change directories to the HP Service Manager WEB-INF folder. For example:

    ```
    <Tomcat>\webapps\webtier_x.xx\WEB-INF
    ```

12. Open the Web configuration file (web.xml) in a text editor.
13. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
```

```
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

14. Save the Web configuration file.
15. Start the Service Manager Web tier.
16. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080
CA certificates file	C:\Program Files\HP\ Service Manager x.xx\Client\plugins\com.hp.ov.sm.client.common_x.xx\cacerts

17. Connect to the Service Manager host.

Requirements for a vertical scaling and required SSL implementation

This configuration is intended for customers who:

- Want to maximize the number of client connections supported on a single host
- Have a host with enough system resources to manage all concurrent client connections
- Want to specify the communications ports the Service Manager implementation uses
- Want to require SSL encryption for all connections
- Want to protect against complex SSL-related attacks
- Want to authenticate that the HP Service Manager server is a valid host

Number of Service Manager hosts required

This implementation requires the following number of hosts.

1

Certificates required

You must create or obtain the following certificates for SSL encryption.

- Certificate authority certificate
- Keystore containing the certificate authority's certificate
- HP Service Manager host certificate

Private keys required

You must create or obtain the following private keys for SSL encryption.

- Certificate authority's private key *
- HP Service Manager host private key

* This key is only necessary if you are managing your own private certificate authority.

Parameters required in sm.cfg

You must set the following configuration parameters.

```
sm -loadbalancer -httpPort:<value>
sm -httpPort:<value> - httpsPort:<value>
sm -httpPort:<value> - httpsPort:<value>
```

- loadbalancer – creates a special servlet container process to route client connection requests to other available servlet container processes
- httpPort – identify the communications port that a servlet container process uses to communicate with clients using HTTP
- httpsPort – identify the communications port that a servlet container process uses to communicate with clients using HTTPS

Parameters required in sm.ini

You must set the following initialization parameters.

- cacertpem – identify the certificate authority's certificate
- certpem – identify the HP Service Manager host's certificate
- pkpem – identify the HP Service Manager host's private key
- pkpempass – identify the password for the HP Service Manager host's private key

- ssl:1
- sessiontimeout – define the number of minutes a client connection can remain unresponsive before the server closes the connection.
- threadsperprocess – identify the total number of threads the servlet container process supports The recommend maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.

Parameters required in web.xml

You must set the following Web parameters.

- cacerts – identify the keystore containing the certificate authority's certificate
- serverHost – identify the host name of the Service Manager host
- serverPort – identify the communications port on which the Service Manager host listens for client connections requests

Windows client preferences required

You must set the following preferences from the **Connection** menu.

- Server host name – identify the host name of the Service Manager host
- Server port number – identify the communications port on which the Service Manager host listens for client connections requests

You must set the following preference from the **Window** > **Preferences** > **HP Service Manager** > **Security** menu.

- CA certificates file – identify the keystore containing the host's certificate authority certificate

Other requirements

You must do the following additional steps to ensure that HP Service Manager can use your private certificates.

- Add the certificate authority's certificate to one or more key stores that your Web and Windows clients can access
- Ensure that the HP Service Manager server's host name matches the common name (CN) listed in the host's signed certificate

Lightweight Directory Access Protocol (LDAP)

You can integrate HP Service Manager to an LDAP directory service to share contact information across your network. Once you have enabled an LDAP integration to HP Service Manager, you can then configure HP Service Manager to automatically create operator records for LDAP users by either of the following methods:

- Defining a user template for LDAP log ins
- Defining a system default record for all log ins

Using either method, you can map fields in the operator record to contact information in the LDAP directory service. This mapping allows HP Service Manager to create an operator record with all the available contact details defined in the LDAP directory service. If you create an LDAP user template, you can make changes to all users built from this template by editing the template operator record. If you create a system default record, then you must manually make changes to each individual operator record that HP Service Manager creates. If you create both an operator template and a system default operator record, HP Service Manager uses the operator template to create new operator records.

Caution: Using the legacy listener with an LDAP integration is NOT supported.

Note: HP Service Manager denies access to LDAP users unless the system administrator defines either an operator template or a system default operator record.

After you have mapped fields in an operator record to a LDAP directory service, only users who are both LDAP administrators and HP Service Manager system administrators can update and add new operator records. HP Service Manager applies this restriction because HP Service Manager synchronizes any changes you make to the operator record with the corresponding LDAP entry. In addition, if you create new operator records, then HP Service Manager also creates new users in the LDAP directory.

Note: Deleting an operator record does not cause HP Service Manager to delete LDAP users. Only an LDAP administrator can delete LDAP entries.

Typically, HP Service Manager system administrators will want to map only the operators file to an LDAP directory, however they can also map any other system table, for example, the contacts or device table, to an LDAP directory. You can map a HP Service Manager table to only one LDAP server at a time, although you may specify a different LDAP server for each table.

When mapping between HP Service Manager and LDAP directories, you can decide which data source you want to be primary. In cases where there are duplicate entries between data sources, HP Service Manager displays only the data listed in the primary data source.

Define file and field-level mappings to an LDAP server

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

You must define a default LDAP server to integrate HP Service Manager files to an LDAP directory service.

To define file and field-level mappings to an LDAP server:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **LDAP Mapping**.
   The HP Service Manager LDAP Mapping – System Level Specification form opens.
2. Click **Set File/Field-level Mappings**.
   The **HP Service Manager LDAP Mapping – File/Field Specification** page opens.
3. In the Name field, type the name of the HP Service Manager file for which you want to create LDAP mappings.
4. Click **Search**.
   HP Service Manager displays a list of fields for the file.
5. Type or select the LDAP mapping settings. If necessary, press Ctrl+H to view help for each field.
6. Click **Save**.
   HP Service Manager displays the message:
   Data Policy record updated.

Define the default LDAP server

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

You must define a default LDAP server to integrate HP Service Manager files to an LDAP directory service.

To define the default LDAP server:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **LDAP Mapping**.
   The HP Service Manager LDAP Mapping – System Level Specification form opens.
2. Type or select the LDAP mapping information.
3. Click **Save**.
   HP Service Manager displays the message;
   Record updated in the scldapconfig file.

Enable an integration to LDAP

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

To enable an integration to LDAP:

1. Do one of the following to enable HP Service Manager to create operator records dynamically when LDAP users log in:
   - Define the System Information Record operator template for LDAP users.
   - Define a system default record for the operator file.

   These settings allow LDAP users to log in to HP Service Manager without having existing operator records.
2. Define the default LDAP server and authentication base directory to which you want HP Service Manager to connect.
3. Define the file and field mappings you want to the LDAP directory service.
4. Set LDAP query parameters in the HP Service Manager initialization file.

Set the LDAP authentication base name

Applies to User Roles:

System Administrator

You can define an operator to use a different LDAP base name than the operator name. By default, HP Service Manager uses the operator name to bind to the LDAP. You can define a different LDAP base name to allow users to connect to HP Service Manager with one name and to LDAP with a different name.

To set the LDAP authentication base name:

1. Click **System Administration** > **Ongoing Maintenance** > **Operators**.
2. Click **Search**.
3. Select the operator whose LDAP base name you want to set from the record list.
4. Click the **Security** tab.
5. In the **LDAP Base Name** field, type the name you want to use to authenticate the LDAP connection.
6. Click **Save**.
   HP Service Manager displays the message:
   Operator record updated.

Enable LDAP over SSL

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use the procedures below.

By default, when you enable LDAP over SSL, you need to set the root certificate of the CA that issued the LDAP server's certificate on the Service Manager server, and then specify the location of the certificate file in the **LDAP SSL DB Path** field.

If you do not want to set the CA's root certificate on the Service Manager server, follow these steps:

1. Set the `ldapsslallownocert` parameter to `1`.
2. Log in to Service Manager, and then click **System Administration** > **Ongoing Maintenance** > **System** > **LDAP Mapping**.
3. Set the **LDAP Server** and **LDAP Base Directory** fields, select the **LDAP SSL** check box. Leave the **LDAP SSL DB Path** field blank.

4. Click **Set File/Field level mapping**, enter `operator` in the **Name** field, and then map the `name` field of `operator` to `sAMAccountName` (for Active Directory server).
5. Restart the Service Manager server.

If you wish to authenticate Service Manager users that belong to different domains or subdomains, you can deploy multiple LDAP servers that belong to the corresponding domains, and then set up a horizontal scaled (HS) cluster. By the following configuration, users belong to different domains can share the same database while at the same time be authenticated by different domain's LDAP server over SSL.

1. Set the `ldapsslallownocert` parameter to `1`.
2. Log in to Service Manager, and then click **System Administration** > **Ongoing Maintenance** > **System** > **LDAP Mapping**. Leave everything on this page empty.
3. Click **Set File/Field Level Mapping**, enter `operator` in the **Name** field, and then map the `name` field of `operator` to `sAMAccountName`.
4. Add the `ldapserver` parameter in the `sm.ini` file as the following example:

   ```
   ldapserver1:16.183.93.217%636%cn=users,dc=swsm,dc=ind,dc=lab
   ```

   You can add this parameter multiple times if you have more than one LDAP server.
5. Restart the Service Manager server.

Note: In both cases above, you still need to set the `ldapbinddn` and `ldapbindpass` parameters in `sm.ini`.

IPv6 overview

As of version 9.32, Service Manager supports IPv4/IPv6 as a dual stack network. This addresses the needs of being able to run Service Manager infrastructure in an organization where both IPv6 and IPv4 are enabled in parallel.

Most Service Manager components support IPv6 from Service Manager 9.32, except for the following legacy features which still support only IPv4:

- SCAuto SDK
- ODBC Driver for Crystal Reports

Note: In a dual stack network, the required IPv4 configuration is the same as before.

Recommended Topologies

We recommend that your implementation follow these topology guidelines:

Topology 1

- Service Manager run-time environment, underlying RDBMS, and all Application/Web Servers in a pure IPv4 environment use IPv4.
- The clients in a pure IPv4 environment using IPv4.

Topology 2

- Service Manager run-time environment, underlying RDBMS, and all Application/Web Servers in a dual IPv4/IPv6 environment use IPv4.
- The clients in pure a IPv4 environment or in a dual IPv4/IPv6 environment both use IPv4.

Topology 3

- Service Manager run-time environment, underlying RDBMS, and all Application/Web Servers in a dual IPv4/IPv6 environment use IPv6.
- The clients in a pure IPv4 environment use IPv4.

- The clients in a dual (IPv4/IPv6) environment use IPv6.

IPv6 supported address formats

IPv6 in Service Manager supports the following IP address formats.

Text representation

Service Manager supports text representation of IPv6 formats as defined in RFC 4291 (IP Version 6 Addressing Architecture).

IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. In general, the standard IPv6 address is *x*:*x*:*x*:*x*:*x*:*x*:*x*:*x* where each *x*represents 1 to 4 hexadecimal digits, as shown in the following example:

```
2001:DB8:0:0:8:800:200C:417A
```

It is not necessary to specify the leading zeros of an individual field. However, you must have at least one hexadecimal character in each field. You can use `::` to represent one or more groups of 16 bits of zeros to avoid specifying the entire IPv6 address. However, you can only use the `::` one time. The `::` may also be used to compress the leading or trailing zeros of the IPv6 address. For example, the following two IPv6 addresses are equivalent:

```
FF01:0:0:0:0:0:0:101
```

```
FF01::101
```

Note: You can use either of these representations when you configure IPv6 for use in Service Manager.

Address representation in mixed IPv4/IPv6 environments

According to the RFC 4291 (IP Version 6 Addressing Architecture) standard, the following format is used to handle mixed IPv4/IPv6 environments:

x:*x*:*x*:*x*:*x*:*x*:*d*.*d*.*d*.*d*

In this case, the *x*s are placeholders for the hexadecimal values of the six high-order 16-bit pieces of the address, and the *d*s are the decimal values of the four low-order 8-bit pieces of the address, which is the standard IPv4 representation. Therefore, you can specify an old IPv4 address in a mixed IPv4/IPv6 environment in either of the following, equivalent ways:

```
0:0:0:0:0:0:13.1.68.3
::13.1.68.3  //Uses :: to minimize the leading zeros of the IPv6 address.
```

Note: In Service Manager this mixed use IPv4/IPv6 is not supported.

IPv6 addresses with a port number

To specify an IPv6 address together with a port number, enclose the IPv6 address in brackets as shown in the following example:

```
[2001:db8::1]:80
```

IPv6 configuration

The following topics provide basic information on how to configure Service Manager to use IPv6:

Configure the Service Manager server for IPv6

To configure the Service Manager server to use IPv6, specify an IPv6 address in the `groupbindaddress` parameter in the `sm.ini` file where the Service Manager server is installed. The following example shows a valid IPv6 address:

```
groupbindaddress:2620:0:a17:e006:1060:cb7b:590a:4388
```

Note: If the `groupbindaddress` parameter is specified with an IPv6 address, Service Manager will listen for both IPv4/IPv6 addresses. If no value is specified, or if an IPv4 address is specified, Service Manager will listen only for IPv4 addresses.

Configure IPv6 for horizontally-scaled environments

Make sure that you configured Service Manager as described in the "Configure the Service Manager server for IPv6" on the previous page section. To configure IPv6 for horizontally scaled environments, you must also specify the following parameters on each horizontally scaled server:

Parameter	Value
`groupbindaddress`	Specify an IPv6 address in the `groupbindaddress` parameter in the `sm.ini` file where the Service Manager server is installed.
`grouplicenseip`	Set as the valid IPv6 or IPv4 address of the machine where the licenses are stored.
`groupmcastaddress`	By default, this parameter is set to `FF02::1` after the `groupbindaddress` parameter is specified. If needed, you can modify this parameter according to the following standard: http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml#ipv6-multicast-addresses-1

Note: All other configurations steps are the same as in an IPv4 environment.

Configure IPv6 for vertically-scaled environments

Make sure that you configured Service Manager as described in the "Configure the Service Manager server for IPv6" on the previous page section. To configure IPv6 for vertically-scaled environments, you must also specify the following variables:

Parameter	Value
`groupbindaddress`	Specify an IPv6 address in the `groupbindaddress` parameter in the `sm.ini` file where the Service Manager server is installed.

Note: All other configurations steps are the same as in an IPv4 environment.

Configure the Windows client for IPv6

To configure the Service Manager Windows client to use IPv6, follow these steps:

1. Navigate to **File** > **Connect..** > **Connections**.
2. In the Connections dialog, enter a valid IPv6 in the "Server host name" field.

Configure the Web clients for IPv6

Application Server

The steps that you use to configure IPv6 depends on the Application server that you are using. Choose the appropriate instructions for your environment.

Tomcat 7.0

The default installation of Tomcat does not support listening on the IPv6 protocol. Therefore, you must download latest APR library (tcnative-1.dll) and add an `address` parameter for the connector in the `server.xml` file. To do this, follow these steps:

1. Download latest APR library as appropriate for your architecture.

Architecture	Download location:
x86	http://archive.apache.org/dist/tomcat/tomcat-connectors/native/1.1.20/binaries/win32/tcnative-1.dll
x64	http://archive.apache.org/dist/tomcat/tomcat-connectors/native/1.1.20/binaries/win64/x64/tcnative-1.dll
IA64	http://archive.apache.org/dist/tomcat/tomcat-connectors/native/1.1.20/binaries/win64/ia64/tcnative-1.dll

2. Replace the default APR library with the one you downloaded.

   ```
   $Tomcat_Installation$\bin\tcnative-1.dll
   ```

3. Add an `address` parameter for the connector in the `server.xml` file as follows:

   ```
   <Connector address="[::]" port="8081" protocol="HTTP/1.1" maxThreads="400"
              connectionTimeout="20000"
              redirectPort="8443" />
   ```

4. Add an `address` parameter for the AJP connector in the `server.xml` file as follows:

   ```
   <Connector address="[::]" port="8008" protocol="AJP/1.3" redirectPort="8443" />
   ```

   Note: Modify the `server.xml` file for each for Tomcat instance. To revert to IPv4, remove `address` parameter in the `server.xml` file.

5. Set `serverHost` parameter of the `WEB-INF\web.xml` to a valid IPv6 address:

   ```
   <param-name>serverHost</param-name>
   <param-value>2620:0:a17:e006:d024:1d09:c2f9:f03f</param-value>
   ```

   Note: You do not need to use enclose the IPv6 address in the brackets ([]).

6. Access the Service Manager Web tier by using a correct URL as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/webtier-9.32/index.do
   ```

WebSphere 7

To configure WebSphere 7 for IPv6, follow these steps:

1. Set `serverHost` parameter of the `WEB-INF\web.xml` to a valid IPv6 address:

   ```
   <param-name>serverHost</param-name>
   <param-value>2620:0:a17:e006:d024:1d09:c2f9:f03f</param-value>
   ```

   Note: You do not need to use enclose the IPv6 address in the brackets ([]).

2. Access the Service Manager Web tier by using a correct URL as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/webtier-9.32/index.do
   ```

WebLogic 11

To configure WebLogic 11 for IPv6, follow these steps:

1. Set `serverHost` parameter of the `WEB-INF\web.xml` to a valid IPv6 address:

   ```
   <param-name>serverHost</param-name>
   <param-value>2620:0:a17:e006:d024:1d09:c2f9:f03f</param-value>
   ```

   Note: You do not need to use enclose the IPv6 address in the brackets ([]).

2. Access the Service Manager Web tier by using a correct URL as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/webtier-9.32/index.do
   ```

JBoss 5.1

To configure JBoss 5.1 for IPv6, follow these steps:

1. Set `serverHost` parameter of the `WEB-INF\web.xml` to a valid IPv6 address:

   ```
   <param-name>serverHost</param-name>
   <param-value>2620:0:a17:e006:d024:1d09:c2f9:f03f</param-value>
   ```

   Note: You do not need to use enclose the IPv6 address in the brackets ([]).

2. Start Jboss an appropriate command similar to the following, together with the IPv6 address in brackets ([]):

   ```
   run.bat -b [2620:0:a17:e006:4a8:91c1:c9cb:93dd]
   ```

3. Access the Service Manager Web tier by using a correct URL as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/webtier-9.32/index.do
   ```

Web Server

The steps that you use to configure IPv6 depends on the Web server that you are using. Choose the appropriate instructions for your environment.

Apache 2.2

To configure Apache to use IPv6, you must download,modify, and recompile Apache. To do this, refer to these example steps:

1. Download and install Perl from the following web site:

   http://www.activestate.com/activeperl/downloads

2. Download and install awk95.exe to some directory, for example, `C:\Windows`. Rename the file to `awk.exe`.

3. Compile zlib. To do this, follow these steps:
   a. Download the source code for zlib from the following web site:

      ```
      http://www.zlib.net/
      ```

   b. Unzip the file to `C:\ipv6\zlib`.

   c. Run the following command to compile zlib:

      ```
      C:\ipv6\zlib\nmake -f win32\Makefile.msc
      ```

d. If you receive an `nmake fatal error code 0xc0000135`, run the following command:

```
C:\ipv6\zlib\vcvars32
```

4. Compile OpenSSL. To do this, follow these steps:
   a. Download the source code for OpenSSL from the following web site:

   ```
   http://www.openssl.org/
   ```

   b. Unzip the file to `C:\ipv6\openssl`.

   c. Run the following command to compile openssl:

   ```
   C:\ipv6\openssl\perl Configure VC-WIN32
   C:\ipv6\openssl\ms\do_ms
   C:\ipv6\openssl\nmake -f ms\ntdll.mak
   ```

5. Compile and modify Apache . To do this, follow these steps:
   a. Download the source code for Apache from the following web site:

   ```
   http://httpd.apache.org/download.cgi
   ```

   b. Unzip the file to `C:\ipv6\httpd`.

   c. Edit the `httpd\srclib\apr\include\apr.hw` file, and change the line `#define APR_HAVE_IPV6 0` to `#define APR_HAVE_IPV6 1`.

   Copy the `zlib` and `openssl` folders from the previous steps to `httpd\srclib`.

   d. Run the following command to compile Apache:

   ```
   C:\ipv6\httpd\nmake /F Makefile.win INSTDIR="C:\Apache" installr
   ```

   Note: You may encounter issues if you try to compile Apache 2,0 or earlier. Specifically, you may receive the following errors:

   ```
   C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\ws2def.h(91) :
   warning C4005: 'AF_IPX' : macro redefinition
           C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\winsock.h
   (460) : see previous definition of 'AF_IPX'
   C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\ws2def.h(127) :
   warning C4005: 'AF_MAX' : macro redefinition
           C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\winsock.h
   (479) : see previous definition of 'AF_MAX'
   C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\ws2def.h(163) :
   ```

```
warning C4005: 'SO_DONTLINGER' : macro redefinition
        C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\winsock.h
(402) : see previous definition of 'SO_DONTLINGER'
C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\ws2def.h(206) :
error C2011: 'sockaddr' : 'struct' type redefinition
        C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\winsock.h
(485) : see declaration of 'sockaddr'
C:\Program Files\Microsoft SDKs\Windows\v6.0A\include\ws2def.h(384) :
error C2059: syntax error : 'constant'
```

These issue are caused by an improper include order between `windows.h` and the `winsock2.h` files in the `apr` library.. To resolve this, make sure that `winsock2.h` is included before `windows.h`.

IIS 7.5

IPv6 is not supported for an IIS connector with Tomcat (mod_jk).

Configure IPv6 to work with Service Manager integrations

Note: All other configurations steps are the same as in an IPv4 environment.

Knowledge Management search server

Make sure that you configured Service Manager as described in the "Configure the Service Manager server for IPv6" on page 93 section. To configure IPv6 in a Knowledge Management search server, follow these steps:

1. In the Service Manager client, navigate to **Knowledge Management** > **Configuration** > **Configure Search Servers**.
2. Set `Hostname` as the IPv6 address of the Knowledge Management search server, including the brackets, as shown:

   `[2620:0:a17:e006:4a8:91c1:c9cb:93dd]`

Note: All other configurations steps are the same as in an IPv4 environment.

LDAP server

Make sure that you configured Service Manager as described in the "Configure the Service Manager server for IPv6" on page 93 section. To configure IPv6 for an LDAP server, follow these steps:

1. In the Service Manager client, navigate to **System Administration** > **Ongoing Maintenance** > **System** > **LDAP Mapping**.
2. Set `LDAP server` as the IPv6 address of the LDAP server, including the brackets, as shown:

   `[2620:0:a17:e006:4a8:91c1:c9cb:93dd]`

Note: All other configurations steps are the same as in an IPv4 environment.

SMTP server

Make sure that you configured Service Manager as described in the "Configure the Service Manager server for IPv6" on page 93 section. To configure IPv6 for an SMTP server, follow these steps:

1. Specify the IPv6 address of the SMTP server in the `smtphost` parameter in the `sm.ini` file. The following example shows a valid IPv6 address:

   `smtphost:2620:0:a17:e006:1060:cb7b:590a:4388`
2. Set the `smtphost` connection parameters for the emailout startup parameter as a valid IPv6 address:

   `sm -emailout -smtphost:[2620:0:a17:e006:4a8:91c1:c9cb:93dd] -smtpport:25`

Note: All other configurations steps are the same as in an IPv4 environment.

Configure IPv6 for SRC, Mobile Applications, and SCAuto

Service Request Catalog

To enable IPv6 support in Service Request Catalog (SRC), follow these steps:

1. Enable IPv6 support in Tomcat. To do this, see the Tomcat section of "Configure the Web clients for IPv6" on page 95.
2. Enable IPv6 in Service Manager. To do this, see "Configure the Service Manager server for IPv6" on page 93.
3. In folder where SRC is deployed, set the `sm.hostname` parameter in the `applicationContext.properties` file to a Full Qualified Domain Name or to a valid IPv6 address

   `sm.hostname=[2620:0:a17:e006:4a8:91c1:c9cb:93dd]`

4. You can access the SRC application by entering the valid IP address in your browser, as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/src-9.32
   ```

Service Manager Mobile Applications

To enable IPv6 support for the Service Manager Mobile Application, follow these steps:

1. Enable IPv6 support in Tomcat. To do this, see the Tomcat section of "Configure the Web clients for IPv6" on page 95.
2. Enable IPv6 in Service Manager. To do this, see "Configure the Service Manager server for IPv6" on page 93.
3. In folder where the mobile web tier is deployed, set theendpoint in the `WEB-INF\web.properties` file to a Full Qualified Domain Name or to a valid IPv6 address

   ```
   endpoint=[2620:0:a17:e006:4a8:91c1:c9cb:93dd]
   ```

4. You can access the Service Manager Mobile Application by entering the valid IP address in your browser, as shown in the following example:

   ```
   http://[2620:0:a17:e006:4a8:91c1:c9cb:93dd]:8080/sm/std
   ```

SCAuto

SCAuto can connect to a Service Manager server by using an IPv4 address in a dual IPv4/IPv6 environment. However, SCAuto cannot function work in a pure IPv6 environment because the SCAuto SDK does not support IPv6.

Print options

HP Service Manager offers list view or detail view print options.

Use Search to display the list of records or a detail record. Follow the instructions below to print in the windows client or the web client:

In the windows client:

1. Click **File** > **Print** or click the **Print** icon.
2. Select one of the following options:
   - **List**: Print the list of records.

     You can specify the **Start** and **Count** fields to print a number of consecutive records in the list.
   - **Detail**: Print a single record.
3. Select the printer and then click **Print**.

In the web client:

1. Do one of the following actions:
   - Print the list of records: Click the **Print** icon on the Record List toolbar or press Ctrl+Alt+P.
   - Print a single record: Click the **Print** icon on the Record Detail toolbar or press Alt+P.
2. Perform one of the following actions in the newly-opened browser window:
   - Right-click the page and then click **Print**.
   - Press Ctrl+P.
3. Select the printer and then click **Print**.

Client-side printing

Users can access the client-side printing feature by using the Print icon. Client-side printing provided by the web browser or Windows operating system is also supported.

Regional settings

Regional settings define local values for month names and abbreviations, date format, system language, and default system currency.

Define the months in the year

Applies to User Roles:

System Administrator

To define the months in the year:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Month Tables** tab.
   The table lists the Full Name and Abbreviation for each month.
3. Make any necessary changes.
4. Click **Save**.

Set the default date format

Applies to User Roles:

System Administrator

To set the default date format:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Date Info** tab.

3. In the **Format** field, type or select one of the following date formats to be the system default.
   - **mm/dd/yy** — standard U.S. date format
   - **dd/mm/yy** — standard European date format
   - **yy/mm/dd** — optional date format
   - **mm/dd/yyyy** — standard U.S. date format
   - **dd/mm/yyyy** — standard European date format
   - **yyyy/mm/dd** — optional date format
4. Click **Save**.

Set the default system currency

Applies to User Roles:

System Administrator

To set the default system currency:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **General** tab.
3. In the **Basis Currency** field, type or select one of the options from the currency file.
4. Click **Save**.

Note: The currency setting in each individual operator record overrides the default system currency setting.

Set the default system language

Applies to User Roles:

System Administrator

Note: The language setting in each individual operator record overrides the default language setting.

To set the default system language:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Misc** tab.
3. In the **Language Code** field, type or select one of the following language options:
   - English
   - French
   - German
   - Italian
   - Japanese
4. Click **Save**.

Servlet implementation

HP Service Manager uses a servlet implementation to manage client connections to the server. The servlet implementation uses a pre-configured embedded Java servlet that does not require any additional installation or configuration of Java components. Administrators can manage the servlet implementation using the HP Service Manager configuration files and management procedures rather than Java interfaces.

A servlet implementation provides the following management features:

- The ability to specify the exact communications ports client connections use
- The ability to specify the maximum number of client connections HP Service Manager will accept
- Industry-standard resilience and scalability options
- Multithread processing capabilities

In a servlet implementation, administrators specify the communications ports the system uses in advance. The total number of communication ports required for a servlet implementation is based on the number of servlet container processes you want the HP Service Manager host to support. Each servlet container process manages a set number of client sessions as determined by the number of threads it contains.

Each servlet container process supports a set number of client connections over two communications ports, one for HTTP communications and one for HTTPS communications. Administrators set the number of client connections per process using the threading parameters. Two communications ports is sufficient for up to approximately 150 client connections (as determined by the HP Service Manager host's memory requirements).

A servlet implementation is ideal for administrators who want to do capacity planning and system resource management. The communications port and threading parameters allow administrators to control the resources that client connections consume. By specifying the communication ports and number of threads available for client connections, administrators can set a limit on the total number of client connections any one hardware resource supports. Setting connection limits prevent client connections from consuming more system resources than desired and allows administrators to set a server load threshold in advance. If a client attempts to connect to a HP Service Manager system that has reached its connection limit, the server refuses the connection and displays an error message that the server is unavailable.

Certain servlet implementations benefit from the native resilience and scalability features of grouping Java servlets. In a horizontal scaling implementation, administrators can create virtual groups where multiple HP Service Manager servers act as a single system. Each member in a virtual group communicates its connection availability and system resource usage to the other members of the group. Should a group member fail, the other group members are unaffected by the outage and the system can route any new client connection requests to another member in the virtual group. If the client connection routing process fails, the existing client connections are unaffected and an administrator can restart the client connection routing process.

Servlet implementation options

The HP Service Manager server supports the following types of servlet implementation options. You may combine these implementation options with other HP Service Manager options to create your own system implementation.

- Single servlet implementation – a single host running a single servlet container implementation
- Vertical scaling implementation – a single host running multiple servlet containers implementation
- Horizontal scaling implementation – multiple hosts running multiple servlet containers implementation

A single servlet implementation does not require a load balancer process. Clients send connection requests directly to the HTTP communications port of the servlet container process. The servlet container's HTTP port becomes the single well-known communications port for the HP Service Manager system. All clients send connection requests to this one communications port. The servlet container process routes each connection request to an available thread in the process. When the number of client connections reaches the number of available threads the system has reached capacity, and the servlet container refuses any additional client connection requests. Each servlet container process can manage approximately one hundred and fifty client connections (as determined by the HP Service Manager host's memory requirements). Implementers can use the threadsperprocess parameter to set the total number of client connections the servlet container process manages.

A vertical scaling implementation requires a load balancer process to route client connection requests to an available servlet container process. Clients send connection requests to the HTTP communications port of the load balancer process. The load balancer's HTTP port becomes the single well-known communications port for the HP Service Manager system. All clients send connection requests to this one communications port. The load balancer process transparently routes client connection requests to any available servlet container processes on the local host. The end-user never sees the change in the communications port number from the load balancer process to the servlet container process. When the number of client connections reaches the total number of available threads on all servlet container

processes the system has reached capacity, and the load balancer refuses any additional client connection requests. Implementers must use the threadsperprocess parameter to set the total number of client connections each servlet container process manages.

A horizontal scaling implementation requires a load balancer process to route client connection requests to servlet containers on multiple hosts. Clients send connection requests to the HTTP communications port of the load balancer process. The load balancer's HTTP port becomes the single well-known communications port for the HP Service Manager system. All clients send connection requests to this one communications port. The load balancer process routes client connection requests to any available servlet container process in the group. When the number of client connections reaches the total number of available threads on all servlet containers on all available hosts in the virtual group, the system has reached capacity, and the load balancer refuses any additional client connection requests. Implementers must use the threadsperprocess parameter to set the total number of client connections each servlet container process manages. Implementers must also set the group parameters and ensure there is a valid network connection among the group members.

Single servlet implementation

In a single servlet implementation you manage all client connections with one multithreaded process.

The key features of this implementation are:

- The HP Service Manager system consists of one HP Service Manager instance running on one physical host
- One dedicated servlet container process manages all client connections
- The implementation allows administrators to specify the communication ports the system uses
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of threadsperprocess

A single servlet implementation is a typically used for development environments or small production environments because it is easy to setup and manage.

You can convert a single servlet implementation into any of the other servlet implementations with the addition of a load balancer process and one or more additional servlet container instances.

Single servlet implementation diagram

The *Single servlet implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running a single servlet container implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Single servlet implementation diagram* is available from the help.

Example: Setting up a single servlet implementation

The following example describes how to set up a single servlet implementation that accomplishes the following:

- Manages all client connections with one multithreaded process
- Supports up to 50 concurrent client connections
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses
- Provides a simple configuration to test servlet features

You can use this example to configure the implementation depicted in the *Single servlet implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\ Service Manager\server\RUN
   ```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following line:

   ```
   sm -httpPort:13081 -httpsPort:13082
   ```

   The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on

which you start the servlet container.
The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

8. Add the following lines:

```
threadsperprocess:50
sessiontimeout:3
```

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.
The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

9. Save the Service Manager initialization file.
10. Start the Service Manager server.
11. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

```
C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
```

12. Open the Web configuration file (web.xml) in a text editor.

13. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

14. Save the Web configuration file.

15. Start the Service Manager Web tier.

16. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13081

17. Connect to the Service Manager host.

Requirements for a single servlet implementation

This configuration is intended for customers who:

- Want to manage all client connections with one multithreaded process
- Only expect up to 50 concurrent client connections
- Want to specify the communications ports the HP Service Manager implementation uses
- Want a simple configuration to test servlet features

Number of Service Manager hosts required

This implementation requires the following number of hosts.
1

Parameters required in sm.cfg

You must set the following configuration parameters.

```
sm -httpPort:<value> - httpsPort:<value>
```

- httpPort – identify the communications port that a servlet container process uses to communicate with clients using HTTP. The servlet container communications port must be unique on the host on which you start the servlet container.
- httpsPort – identify the communications port that a servlet container process uses to communicate with clients using HTTPS. The servlet container communications port must be unique on the host on which you start the servlet container.

Parameters required in sm.ini

You must set the following initialization parameters.

- sessiontimeout – define the number of minutes a client connection can remain unresponsive before the server closes the connection.

- threadsperprocess – identify the total number of threads the servlet container process supports. Use a value of threads that maximizes the system resources of your Service Manager host.
  The recommend maximum value for the parameter threads per process is 60. Usually the value of this parameter should be below 50.

Parameters required in web.xml

You must set the following Web parameters.

- serverHost – identify the host name of the Service Manager host
- serverPort – identify the communications port on which the Service Manager host listens for client connections requests

Windows client preferences required

You must set the following preferences from the **Connection** menu.

- Server host name – identify the host name of the Service Manager host
- Server port number – identify the communications port on which the Service Manager host listens for client connections requests

Vertical scaling implementation

In a vertical scaling implementation you maximize the number of client connections supported on a single host.

The key features of this implementation are:

- The HP Service Manager system consists of several servlet container processes running on one physical host
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threadsperprocess value (For example, 6 servlet container processes supporting 50 threadsperprocess can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses

- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove HP Service Manager instances from a virtual group

A vertical scaling implementation is typically used in small to medium environments where hardware system resources are limited. A vertical scaling implementation can support as many client connections as the HP Service Manager host has available system resources.

You can convert a vertical scaling implementation into a horizontal scaling implementations with the addition of virtual grouping parameters and one or more additional servlet container instances installed on separate physical hosts.

Note: There can be only **one** kmupdate process running at any time regardless of the number of hosts. Starting more than one kmupdate process causes unpredictable behavior on the search engine server.

Vertical scaling implementation diagram

The *Vertical scaling implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running multiple servlet containers implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Vertical scaling implementation diagram* is available from the help.

Example: Setting up a vertical scaling implementation

The following example describes how to set up a vertical scaling implementation that accomplishes the following:

- Maximizes the number of client connections supported on a single host
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses

You can use this example to configure the implementation depicted in the *Vertical scaling implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the HP Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\ Service Manager\server\RUN
   ```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following lines:

   ```
   sm -loadBalancer -httpPort:13080
   sm -httpPort:13081 -httpsPort:13082
   sm -httpPort:13083 -httpsPort:13084
   ```

   The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A vertical scaling implementation only needs one load balancer.
   The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.
   The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.
   The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.
   The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.
   The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Verify that the auth parameter has a valid authorization code.
8. Add the RDBMS connection settings. For example:

   ```
   [oracle10]
   sqldb:ora102
   ```

```
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

9. Add the following lines:

   ```
   threadsperprocess:50
   sessiontimeout:3
   ```

   The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

   The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

10. Save the Service Manager initialization file.

11. Start the Service Manager server.

12. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

    ```
    C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
    ```

13. Open the Web configuration file (web.xml) in a text editor.

14. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

15. Save the Web configuration file.
16. Start the Service Manager Web tier.
17. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080

18. Connect to the Service Manager host.

Configuring a vertical scaling environment

This configuration is intended for customers who:

- Want to maximize the number of client connections supported on a single host
- Have a host with enough system resources to manage all concurrent client connections

Terminology

- **Group**: A group of Service Manager server processes running on one or more hosts and connecting to one database to serve Service Manager server processes within a group.
- **Node**: A Service Manager server process within a group.
- **Service Manager servlet process**: A Service Manager server process that embeds the Tomcat application server and serves Service Manager Windows clients and Web clients, as well as Web Services requests. It is also generally referred to as a Service Manager process.
- **Service Manager background process**: Background processes that wake up periodically to execute a particular RAD application, or Service Manager runtime environment routine. For example: `sm -que:ir`.
- **Service Manager transient process**: A Service Manager server process that executes a RAD application or Service Manager runtime environment routine only once, not periodically. For example: `sm -reportlbstatus`.

Commands required in sm.cfg

You must set the following configuration commands.

```
sm -loadBalancer -httpPort:<value>
sm -httpPort:<value> -httpsPort:<value>
sm system.start
```

- sm -loadBalancer -httpPort:<value> – Is a special Service Manager servlet process that redirects client connections requests to other available Service Manager processes.
- sm -httpPort:<value> -httpsPort:<value> – Specifies the HTTP or HTTPS communications port for one Service Manager servlet process. Each Service Manager process by default can host 50 users. You can start as many Service Manager servlet processes as you want, as long as your server has enough physical memory and resources to start the processes. You can also change the number of users hosted on one Service Manager servlet process by setting the "threadsperprocess" parameter.

Optional parameters in sm.ini

The following initialization parameters are optional.

```
threadsperprocess:<value>
preferredFQHN:<value>
groupbindaddress:<value>
```

- threadsperprocess – Defines the maximum number of concurrent user sessions per Service Manager servlet process. Use a value that maximizes the system resources of your Service Manager host. The recommended maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.
- preferredFQHN – Specifies the fully qualified host name you want Service Manager clients to use when communicating with the server. Service Manager loadBalancer redirects client requests to the target host with the target host's "preferredFQHN." You only need to set this parameter if your Service Manager host is identified by multiple names in the network.
- groupbindaddress – Defines the TCP/IP address of the network adapter you want Service Manager processes to use to communicate with other processes in the group.

Memory requirements

HP recommends around 1.5 GB RAM per Service Manager process. This is based on the expected user load. For example, if you have 10 processes each running with a 50-user load, the minimum required RAM would be 1.5 GB per Service Manager process, 10x1.5 GB RAM. For information on sizing, refer to the *Service Manager 7 Reference Configurations* sizing guide in the HP Software online support knowledge documents at the following URL: www.hp.com/go/hpsoftwaresupport.

Licensing requirements

Obtain an AutoPass license for your vertical scaling environment. On Windows platforms, AutoPass installs as part of the server installation. On Unix platforms, you must install it manually before you can run Service Manager. For more information, see the *HP Service Manager Installation Guide* in the related topics. AutoPass includes this host's IP address in the `LicFile.txt` file.

Do the following:

- Get the license from the HP Web site.
- Copy the primary server's IP address to get the license and save it as the primary server's `LicFile.txt` file. The default directory for this file is as follows:
  - **On Windows**: C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data\LicFile.txt
  - **On Unix**: /var/opt/OV/HPOvLIC/LicFile.txt

Vertical scaling and required SSL implementation

In a vertical scaling and required SSL implementation you maximize the number of client connections supported on a single host and provided SSL-encrypted security between client and server

communications.

The key features of this implementation are:

- The HP Service Manager system consists of several servlet container processes running on one physical host
- The implementation allows administrators to specify a system connection limit
- The HP Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threadsperprocess value (For example, 6 servlet container processes supporting 50 threadsperprocess can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses
- The implementation requires SSL-encrypted communications between HP Service Manager clients and servers
- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove HP Service Manager instances from a virtual group

A vertical scaling implementation is typically used in small to medium environments where hardware system resources are limited. A vertical scaling implementation can support as many client connections as the HP Service Manager host has available system resources.

You can convert a vertical scaling implementation into a horizontal scaling implementations with the addition of virtual grouping parameters and one or more additional servlet container instances installed on separate physical hosts.

Vertical scaling and required SSL implementation diagram

The *Vertical scaling and required SSL implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for a single server running multiple servlet containers in an required SSL implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Vertical scaling and required SSL implementation diagram* is available from the help.

Example: Setting up a vertical scaling and required SSL implementation

The following example describes how to set up a vertical scaling and required SSL implementation that accomplishes the following:

- Maximizes the number of client connections supported on a single host
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses
- Require SSL encryption for all connections
- Protects against complex SSL-related attacks
- Authenticates that the Service Manager server is a valid host

You can use this example to configure the implementation depicted in the *Vertical scaling and required SSL implementation diagram*.

1. Install Service Manager on one host. For example: myserver1.
2. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\Service Manager x.xx\Server\RUN
   ```

3. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.
4. Add the following lines:

   ```
   sm -loadBalancer -httpPort:13080
   sm -httpPort:13081 -httpsPort:13082
   sm -httpPort:13083 -httpsPort:13084
   ```

   - The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A vertical scaling implementation only needs one load balancer.
   - The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.

- The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.
- The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.
- The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.
- The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

5. Save the Service Manager configuration file.
6. Open the Service Manager initialization file (sm.ini) in a text editor.
7. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

8. Add the following lines:

```
threadsperprocess:50
sessiontimeout:3
truststoreFile:cacert.keystore
truststorePass:<cacert password>
keystoreFile:scserver.keystore
keystorePass:<server certificate password>
ssl:1
```

- The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

- The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.
- The truststoreFile:cacert.keystore parameter defines the file name and path to the keystore containing a list of trusted CA certificates. This value assumes you are using the default trust store file provided in the Service Manager server's RUN folder. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to find the trust store file.
- The truststorePass:<cacert password> parameter specifies the password to the trust store file. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to read the password value.
- The keystoreFile:scserver.keystore parameter defines the file name and path to the keystore containing the server's certificate file and private key. This value assumes you are using the default trust store file provided in the Service Manager server's RUN folder. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to find the keystore file.
- The keystorePass:<server certificate password> parameter specifies the password to the keystore file. This parameter must not be encrypted with the new initialization parameter encryption feature or the Java components will not be able to read the password value.
- The ssl:1 parameter requires the Service Manager server to use a signed server certificate for SSL-encryption of all client-server communications. Each client connection validates the server's certificate against the signing certificate authority. You must also use the keystoreFile and keystorePass parameters to define the location of the server certificate and private key.

9. Save the Service Manager initialization file.
10. Start the Service Manager server.
11. Log in to the operating system of the HP Service Manager Web tier host and change directories to the HP Service Manager WEB-INF folder. For example:

    ```
    <Tomcat>\webapps\webtier_x.xx\WEB-INF
    ```

12. Open the Web configuration file (web.xml) in a text editor.

13. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

14. Save the Web configuration file.

15. Start the Service Manager Web tier.

16. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080
CA certificates file	C:\Program Files\HP\ Service Manager x.xx\Client\plugins\com.hp.ov.sm.client.common_x.xx\cacerts

17. Connect to the Service Manager host.

Requirements for a vertical scaling and required SSL implementation

This configuration is intended for customers who:

- Want to maximize the number of client connections supported on a single host
- Have a host with enough system resources to manage all concurrent client connections
- Want to specify the communications ports the Service Manager implementation uses
- Want to require SSL encryption for all connections
- Want to protect against complex SSL-related attacks
- Want to authenticate that the HP Service Manager server is a valid host

Number of Service Manager hosts required

This implementation requires the following number of hosts.

1

Certificates required

You must create or obtain the following certificates for SSL encryption.

- Certificate authority certificate
- Keystore containing the certificate authority's certificate
- HP Service Manager host certificate

Private keys required

You must create or obtain the following private keys for SSL encryption.

- Certificate authority's private key *
- HP Service Manager host private key

* This key is only necessary if you are managing your own private certificate authority.

Parameters required in sm.cfg

You must set the following configuration parameters.

```
sm -loadbalancer -httpPort:<value>
sm -httpPort:<value> - httpsPort:<value>
sm -httpPort:<value> - httpsPort:<value>
```

- loadbalancer – creates a special servlet container process to route client connection requests to other available servlet container processes
- httpPort – identify the communications port that a servlet container process uses to communicate with clients using HTTP
- httpsPort – identify the communications port that a servlet container process uses to communicate with clients using HTTPS

Parameters required in sm.ini

You must set the following initialization parameters.

- cacertpem – identify the certificate authority's certificate
- certpem – identify the HP Service Manager host's certificate
- pkpem – identify the HP Service Manager host's private key
- pkpempass – identify the password for the HP Service Manager host's private key
- ssl:1
- sessiontimeout – define the number of minutes a client connection can remain unresponsive before the server closes the connection.
- threadsperprocess – identify the total number of threads the servlet container process supports The recommend maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.

Parameters required in web.xml

You must set the following Web parameters.

- cacerts – identify the keystore containing the certificate authority's certificate
- serverHost – identify the host name of the Service Manager host
- serverPort – identify the communications port on which the Service Manager host listens for client connections requests

Windows client preferences required

You must set the following preferences from the **Connection** menu.

- Server host name – identify the host name of the Service Manager host
- Server port number – identify the communications port on which the Service Manager host listens for client connections requests

You must set the following preference from the **Window** > **Preferences** > **HP Service Manager** > **Security** menu.

- CA certificates file – identify the keystore containing the host's certificate authority certificate

Other requirements

You must do the following additional steps to ensure that HP Service Manager can use your private certificates.

- Add the certificate authority's certificate to one or more key stores that your Web and Windows clients can access
- Ensure that the HP Service Manager server's host name matches the common name (CN) listed in the host's signed certificate

Horizontal scaling implementation

In a horizontal scaling implementation you maximize the number of client connections supported across multiple hosts.

The key features of this implementation are:

- The HP Service Manager system consists of multiple Service Manager servlet container processes running on separate physical hosts
- The implementation allows administrators to specify a system connection limit

- The Service Manager system can manage a number of concurrent client connections up to the number of servlet container processes times the threads per process value (For example, 6 servlet container processes supporting 50 threads per process can support up to 300 client connections)
- The implementation allows administrators to specify the communication ports the system uses
- A dedicated load balancer process manages and routes client connections to available servlet container processes
- Administrators can dynamically add and remove Service Manager instances from a virtual group

A horizontal scaling implementation is typically used in large 24 by 7 environments where system scalability and resilience is a concern. A horizontal scaling implementation can support as many client connections as the sum of the individual Service Manager instances can support (for example, if each instance can support 50 client connections and there are 6 instances then the total system can support 300 client connections). A horizontal scaling implementation has improved resilience features such as the ability to stop and start the load balancer process without causing a total system outage and the ability to add a new servlet container process to a virtual group while the system is running.

Each host in a horizontal scaling implementation can also support a vertical scaling implementation with the addition of the appropriate servlet implementation parameters. This allows each Service Manager host to support multiple servlet container processes and to make the best use of the available system resources.

Note: There can be only **one** kmupdate process running at any time regardless of the number of hosts. Starting more than one kmupdate process causes unpredictable behavior on the search engine server.

Horizontal scaling implementation diagram

The *Horizontal scaling implementation diagram* is a reference diagram for implementers who are responsible for installing and configuring HP Service Manager. The guide shows the configuration settings and network connections required for multiple servers running multiple servlet containers implementation.

You can view and search this guide using Adobe® Reader, which you can download from the Adobe Web site.

The *Horizontal scaling implementation diagram* is available from the help.

Example: Setting up a horizontal scaling implementation

The following example describes how to set up a horizontal scaling implementation that accomplishes the following:

- Maximizes the number of client connections supported across multiple hosts
- Allows an administrator to specify the communications ports the HP Service Manager implementation uses

You can use this example to configure the implementation depicted in the *Horizontal scaling implementation diagram*.

1. Install Service Manager on the first host in the virtual group. For example: myserver1.
   The host must be on the same subnet as the other hosts in the virtual group.
2. Log in to the Webware software licensing center and obtain a license file for this host.

   Note: After you obtain a license file for this host, it becomes the primary host of the horizontal scaling implementation. You must start this host first when starting a horizontal scaling implementation for the first time.
3. Copy the license file you obtain onto a network share accessible to the other servers that will make up the horizontal scaling implementation.
4. Log in to the operating system of the primary host and change directories to the server's AutoPass folder. For example:

   ```
   C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
   ```
5. Paste the license file (LicFile.txt) into the AutoPass folder.
6. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

   ```
   C:\Program Files\HP\Service Manager\server\RUN
   ```
7. Open the Service Manager configuration file (sm.cfg) in a text editor.
8. Edit the file so that only the following lines appear:

   ```
   sm -loadBalancer -httpPort:13080
   sm -que:ir
   ```

```
sm -httpPort:13081 -httpsPort:13082
sm -httpPort:13083 -httpsPort:13084
sm system.start
```

The loadBalancer parameter creates a special servlet container process to route client connection requests to other available servlet container processes. A horizontal scaling implementation only needs one load balancer.

The httpPort:13080 parameter specifies that the load balancer process listens to client connection requests on HTTP port 13080. This communications port must be unique across all hosts that you want to join the Service Manager virtual group.

The que:ir parameter starts the processing of scheduled IR records that the server generates running in asynchronous IR mode.

The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The system.start parameter specifies that the host system should start all the background processes in the system.start script file. This host will be the only host in the virtual group that runs the background processes. The configuration file should already list this line.

9. Save the Service Manager configuration file.
10. Open the Service Manager initialization file (sm.ini) in a text editor.
11. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
```

```
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

12. Add the following lines:

```
grouplicenseip:10.0.0.135
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupsubnetaddress:255.255.255.0
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The communications port can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

13. Save the Service Manager initialization file.

14. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

    ```
    127.0.0.1 localhost
    127.0.0.2 myserver2.mydomain.com myserver2 loghost
    127.0.0.3 myserver3.mydomain.com myserver3 loghost
    ```

    Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

15. Start the Service Manager server.

16. Install Service Manager on the next host in the virtual group. For example: myserver2.
    The host must be on the same subnet as the other hosts in the virtual group.

17. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

    ```
    C:\Program Files\HP\Service Manager\server\RUN
    ```

18. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.

19. Edit the file so that only the following lines appear:

```
sm -httpPort:13081 -httpsPort:13082
sm -httpPort:13083 -httpsPort:13084
sm -sync
```

Note: You do not need the system.start line on this host as only one host in the virtual group needs to run the background processes.

The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The sync parameter specifies that the host system should start the sync background process to identify and remove unused processes.

20. Save the Service Manager configuration file.
21. Open the Service Manager initialization file (sm.ini) in a text editor.
22. Add the RDBMS connection settings. For example:

    ```
    [oracle10]
    sqldb:ora102
    sqllogin:sm7user/password
    sqllibrary:SQORACLE.OCI10.DLL
    sqldictionary:oracle10
    ```

23. Add the following lines:

    ```
    grouplicenseip:10.0.0.135
    groupname:mycluster1
    groupmcastaddress:224.0.1.255
    groupsubnetaddress:255.255.255.0
    ```

```
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The TCP/IP address can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

24. Save the Service Manager initialization file.

25. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

```
127.0.0.1 localhost
127.0.0.2 myserver2.mydomain.com myserver2 loghost
127.0.0.3 myserver3.mydomain.com myserver3 loghost
```

Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

26. Log in to the operating system of the Service Manager server host and copy the primary host's license file (LicFile.txt) from the network share.

27. Change directories to this server's AutoPass folder. For example:

```
C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
```

28. Paste the primary host's license file into this server's AutoPass folder.

29. Start the Service Manager server.

Note: The primary host must be running in order for this host to validate the license file and start up.

30. Install Service Manager 9.41 on the next host in the virtual group. For example: myserver3.
The host must be on the same subnet as the other hosts in the virtual group.

31. Log in to the operating system of the Service Manager server host and change directories to the Service Manager RUN folder. For example:

```
C:\Program Files\HP\Service Manager 9.41\server\RUN
```

32. Open the Service Manager configuration file (sm.cfg or smstart) in a text editor.

33. Edit the file so that only the following lines appear:

```
sm -httpPort:13081 -httpsPort:13082
sm -httpPort:13083 -httpsPort:13084
sm -sync
```

Note: You do not need the system.start line on this host as only one host in the virtual group needs to run the background processes.

The httpPort:13081 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13081. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13082 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13082. This communications port must be unique on the host on which you start the servlet container.

The httpPort:13083 parameter specifies that the servlet container process listens to client connection requests on HTTP port 13083. This communications port must be unique on the host on which you start the servlet container.

The httpsPort:13084 parameter specifies that the servlet container process listens to client connection requests on HTTPS port 13084. This communications port must be unique on the host on which you start the servlet container.

The sync parameter specifies that the host system should start the sync background process to identify and remove unused processes.

34. Save the Service Manager configuration file.

35. Open the Service Manager initialization file (sm.ini) in a text editor.

36. Add the RDBMS connection settings. For example:

```
[oracle10]
sqldb:ora102
sqllogin:sm7user/password
sqllibrary:SQORACLE.OCI10.DLL
sqldictionary:oracle10
```

37. Add the following lines:

```
grouplicenseip:10.0.0.135
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupsubnetaddress:255.255.255.0
```

```
groupport:13100
threadsperprocess:50
sessiontimeout:3
system:13080
ir_asynchronous:1
```

Note: This example assumes the host system has only one network adapter card and therefore does not need the groupbindaddress parameter. If your host has multiple network adapter cards, you must add the groupbindaddress parameter to specify which network adapter the virtual group will use for communications.

The grouplicenseip:10.0.0.135 parameter defines the TCP/IP address of the Service Manager host with a valid AutoPass license for the virtual group. The value of this parameter must match the IP address specified in the AutoPass license.

The groupname:mygroup1 parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. The value is arbitrary and can be any text value without spaces.

The groupmcastaddress:244.0.1.255 parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process. The TCP/IP address must be consistent with the UDP multicasting protocol.

The groupsubnetaddress:255.255.255.0 parameter defines the subnet mask that servlet container processes can use to communicate with the load balancer process. The subnet mask must be consistent with the IPv4 protocol.

Note: You only have to define the subnet address if you are using subnet masking on your IP addresses.

The groupport:13100 parameter defines the communications port that servlet container processes can use to communicate with the load balancer process. The TCP/IP address can be any available communications port.

The threadsperprocess:50 parameter defines the number of threads each process supports. A value of 50 threads assumes that the Service Manager host has the minimum recommended system memory available for servlet container processes.

The sessiontimeout:3 parameter defines the number of minutes that the client connection can remain unresponsive before the server closes the connection. A value of 3 minutes assumes that most network latency issues are quickly and easily resolved.

The system:13080 parameter defines a numerical identifier for the system, in this case the HTTP port of the load balancer process. The value of this parameter must match for each system in the virtual group.

The ir_asynchronous:1 parameter defines how the server generates IR index files. A value of 1 means that the server creates a schedule record to process the files asynchronously.

38. Save the Service Manager initialization file.

39. If the host runs on a Unix operating system and uses an SSL implementation, edit the /etc/hosts file to list the fully qualified domain name of each host in the virtual group. For example:

```
127.0.0.1 localhost
127.0.0.2 myserver2.mydomain.com myserver2 loghost
127.0.0.3 myserver3.mydomain.com myserver3 loghost
```

Caution: Do not edit the /etc/hosts file unless your system is running an SSL implementation.

40. Log in to the operating system of the Service Manager server host and copy the primary host's license file (LicFile.txt) from the network share.

41. Change directories to this server's AutoPass folder. For example:

```
C:\Program Files\Common Files\Hewlett-Packard\HPOvLIC\data
```

42. Paste the primary host's license file into this server's AutoPass folder.

43. Start the Service Manager server.

Note: The primary host must be running in order for this host to validate the license file and start up.

44. Log in to the operating system of the Service Manager Web tier host and change directories to the Service Manager WEB-INF folder. For example:

```
C:\apache-tomcat-5.5.12\webapps\sm\WEB-INF
```

45. Open the Web configuration file (web.xml) in a text editor.

46. Set the following parameter values:

Parameter	Default value	Description
secureLogin	true	Controls the encryption of network communication between the web

Parameter	Default value	Description
		application server and the web browser. Set it to false if you do not use Secure Sockets Layer (SSL) connections to the web server. **Note**: To use secure login, you must enable SSL on your web application server. For details, refer to your web application server documentation.
sslPort	8443	This parameter is needed only when secureLogin is set to true. Set it to the SSL port of the web application server.
serverHost	localhost	Specifies the name of the Service Manager host server.
serverPort	13080	Specifies the communications port number to which the Service Manager server listens.

```
...
<context-param>
  <param-name>secureLogin</param-name>
  <param-value>true</param-value>
</context-param>
<context-param>
  <param-name>sslPort</param-name>
  <param-value>8443</param-value>
</context-param>
...
  <param-name>serverHost</param-name>
  <param-value>myserver1</param-value>
</init-param>
<init-param>
  <param-name>serverPort</param-name>
  <param-value>13080</param-value>
</init-param>
```

47. Save the Web configuration file.
48. Start the Service Manager Web tier.
49. Open Service Manager Windows clients and set the Service Manager host name and communication port values:

Field	Value
Server host name	myserver1
Server port number	13080

50. Connect to the Service Manager host.

Configuring a horizontal scaling environment

This configuration is intended for customers who:

- Want to maximize the number of client connections supported across multiple hosts
- Have multiple hosts available to manage all concurrent client connections

Terminology

- **Group**: A group of Service Manager server processes running on one or more hosts and connecting to one database to serve Service Manager server processes within a group.
- **Node**: A Service Manager server process within a group.
- **Primary host**: The host in the Service Manager horizontal scaling group whose IP is bound to the Service Manager AutoPass license.
- **Secondary host**: The host in the Service Manager horizontal scaling group whose IP is not bound to the Service Manager AutoPass license. The Service Manager processes on this host depend on running Service Manager processes on the primary host to validate the Service Manager AutoPass license.
- **Service Manager servlet process**: A Service Manager server process that embeds the Tomcat application server and serves Service Manager Windows clients and Web clients, as well as Web Services requests. It is also generally referred to as a Service Manager process.
- **Service Manager background process**: Background processes that wake up periodically to execute a particular RAD application, or Service Manager runtime environment routine. For example: `sm -que:ir`.
- **Service Manager transient process**: A Service Manager server process that executes a RAD application or Service Manager runtime environment routine only once, not periodically. For example: `sm -reportlbstatus`.

Number of Service Manager hosts required

This configuration can be set up on one or more hosts.

Commands required in sm.cfg on primary host

You must set the following configuration commands.

```
sm -loadBalancer -httpPort:<value>
sm -que:ir
sm -httpPort:<value> -httpsPort:<value>
sm system.start
```

Commands required in sm.cfg on secondary host

```
sm -httpPort:<value> -httpsPort:<value>
sm -sync
```

- sm -loadBalancer -httpPort:<value> – Is a special Service Manager process that redirects client connections requests to other available Service Manager processes.
- sm -httpPort:<value> -httpsPort:<value> – Specifies the HTTP or HTTPS communications port for one Service Manager process. Each Service Manager process by default can host 50 users. You can start as many Service Manager processes as you want, as long as your server has enough physical memory and resources to start the processes. You can also change the number of users hosted on one Service Manager process by setting the "threadsperprocess" parameter.
- sm -que:ir – Starts the processing of scheduled IR records that the server generates running in asynchronous IR mode. Since asynchronous IR mode is enforced in a horizontal scaling environment, there has to be only one "sm -que:ir" process for the horizontal scaling group.
- sm system.start – Starts one Service Manager process that contains all background processes.
- sm -sync – Identifies defunct Service Manager processes on the host and frees the shared resources that were allocated for it. The "sm -sync" process needs to start only one per host. Since the "sm system.start" process includes the "sm -sync" background process, the "sm -sync" command should not be included in the `sm.cfg` file that has the "sm system.start" command.

Parameters required in sm.ini

You must set the following initialization parameters.

```
system:<value>
grouplicenseip:<value>
groupname:<value>
groupmcastaddress:<value>
groupport:<value>
```

- system – Defines a unique numerical ID for the system. The value of this parameter must be identical in the `sm.ini` files for each host in the horizontal scaling group.
- grouplicenseip – The value should be the primary host IP.
- groupname – Creates a group name as the horizontal scaling group identifier that the Service Manager processes use to identify the group. The value of this parameter is only alphanumeric characters.
- groupmcastaddress – Defines the TCP/IP multicast address that all Service Manager processes within the horizontal scaling group use to communicate with each other.
- groupport – Defines the communications port that all Service Manager processes use to communicate with all other Service Manager processes within the horizontal scaling group.
- [RDBMS Settings] – Define the RDBMS connection and authorization parameters.

Optional parameters in sm.ini

The following initialization parameters are optional.

```
threadsperprocess:<value>
preferredFQHN:<value>
groupbindaddress:<value>
```

- threadsperprocess – Defines the maximum number of concurrent user sessions per Service Manager process. Use a value that maximizes the system resources of your Service Manager host. The recommended maximum value for the parameter threadsperprocess is 60. Usually the value of this parameter should be below 50.
- preferredFQHN – Specifies the fully qualified host name you want Service Manager clients to use when communicating with the server. Service Manager loadBalancer redirects client requests to the target host with the target host's "preferredFQHN." You only need to set this parameter if your Service Manager host is identified by multiple names in the network.
- groupbindaddress – Defines the TCP/IP address of the network adapter you want Service Manager processes to use to communicate with other processes in a horizontal scaling group. If your Service Manager hosts contain multiple network adapters, you must specify the IP address of the network adapter you want the horizontal scaling group to use with the "groupbindaddress" parameter.

Operating system requirements

All hosts in the horizontally-scaled environment must run on the same operating system version and run at the same patch level. Minimum patch level for each operating system is specified in the support matrix. See HP Support Matrices" on the Software Support Online site.

Caution: Running Service Manager in a horizontally-scaled environment and using different operating systems can corrupt your data. You can also run into problems when upgrading to the latest patches. Most of the time, Service Manager patches are released on all operating systems. However, in some cases, patches may be delivered to a particular operating system. When mixing operating systems, applying patches will become more complex as you need to test on multiple platforms.

Network requirements

All hosts in the horizontal scaling group must run from the same subnet on the network in order for the Service Manager processes to communicate with one another.

Memory requirements

HP recommends around 1.5 GB RAM per Service Manager process. This is based on the expected user load. For example, if you have 10 processes each running with a 50-user load, the minimum required RAM would be 1.5 GB per Service Manager process, 10x1.5 GB RAM. For information on sizing, refer to the *Service Manager 7 Reference Configurations* sizing guide in the HP Software online support knowledge documents at the following URL: www.hp.com/go/hpsoftwaresupport.

Licensing requirements

Obtain an AutoPass license for one host in your horizontal scaling group. On Windows platforms, AutoPass installs as part of the server installation. On Unix platforms, you must install it manually before you can run Service Manager. For more information, see the *HP Service Manager Installation Guide* in the related topics. This host becomes the primary host of the horizontal scaling group and AutoPass includes this host's IP address in the `LicFile.txt` file. You must start the primary host first when starting the horizontal scaling group. Copy the `LicFile.txt` file from the primary host's AutoPass directory to the AutoPass directory of each secondary host in the horizontal scaling group. Each secondary host must have a copy of the primary host's `LicFile.txt` file in order to start.

The summary steps are as follows:

- List all hosts in your horizontal scaling environment to determine the primary server and all secondary servers.
- Get the license from the HP Web site.
- Copy the primary server's IP address to get the license and save it as the primary server's `LicFile.txt` file.

- Copy the primary server's `LicFile.txt` file on all secondary servers. The default directory for this file is as follows:

  `<Service Manager server installation path>/RUN/LicFile.txt`

- Add the following parameter in the `sm.ini` file: `grouplicenseip:Primary_Servers_IpAddress`

To start the horizontal scaling group

1. Start the Service Manager server on the primary server.

   Note: The primary server must be started prior to starting any of the secondary servers.

2. Start the Service Manager server on the secondary servers, in no particular order.

To establish the horizontal scaling group, the first node of the group should be started on the primary host. When the group is established on the primary host, Service Manager server processes on secondary hosts can start to join the group. Once the whole group is established, the primary host can be brought down for maintenance, and then rejoin the rest of the running group when it is restarted. As long as there is a member Service Manager server process running in the group, Service Manager server processes on another secondary host with the same "grouplicenseip:<primary host IP>" can join the group. The primary host is not required to be running. Another secondary host can be started, even if the primary host is down for any reason. However, when the group is down (for example, there are no nodes running in the group), then the group has to be reestablished from the primary host first.

Other requirements

Install the applicable database client software on each host of the horizontal scaling group so that they can access the Service Manager RDBMS. See your RDBMS vendor documentation for instructions.

Servlet implementation processes

All HP Service Manager servlet implementations use multithreaded processes. To control the client connections to these threads, you can configure the HP Service Manager server to start one of two types of processes:

- One or more servlet container processes that accept direct HTTP and HTTPS client connections
- One load balancer process to be the master thread controller for all servlet container processes in a virtual group

A servlet container process is a HP Service Manager sm.exe process that includes a multithreaded Java servlet container. Each servlet container process manages a pool of child threads dedicated to handling HP Service Manager client transactions. The Servlet container process listens for incoming client connection requests on a predetermined HTTP port and routes the requests to an available thread in the process. The thread then binds to the HTTP or HTTPS port of the parent servlet container process for the duration of the client connection. The total number of threads available determine the total number of client connections the servlet container process can manage. Each active thread in the servlet container process holds system locks and communicates this information to other servlet container processes in the same virtual group.

A load balancer process is an HP Service Manager sm.exe process dedicated to routing incoming client requests to servlet container processes in vertical and horizontal scaling implementations. In either scaling implementation the virtual group only requires one load balancer process. The Load balancer process does not have any child threads of its own, nor does it maintain any system locks or hold system resources. The load balancer listens for incoming client connection requests on a predetermined HTTP port and routes them to an available servlet container. The available servlet container may be on the same physical system or on another system in the same virtual group depending on the implementation.

Property	Servlet container process	Load balancer process
Routes client connections to servlet container processes?	No	Yes
Routes client connections to threads?	Yes	No
Creates child threads?	Yes	No
Accepts client connections?	Yes	No
Uses an HTTP port?	Yes	Yes
Uses an HTTPS port?	Yes (optional)	No

Parameter: groupmcastaddress

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

groupmcastaddress

Description

This parameter defines the TCP/IP address that servlet container processes can use to communicate with the load balancer process in a horizontal scaling implementation. The servlet container processes talk to one another using User Datagram Protocol (UDP) multicasting and must use a TCP/IP address consistent with that protocol. You must enable UDP multicasting traffic on your network to use HP Service Manager virtual grouping.

It is best practice to place this parameter in the HP Service Manager initialization file so that all servlet container processes started on the same host share the TCP/IP address specified by this parameter.

Valid if set from

Server's operating system command prompt

Initialization file (sm.ini)

Requires restart of the Service Manager server?

Yes

Default value

None

Possible values

Any TCP/IP address valid for UDP multicasting (addresses 224.0.1.0 to 239.255.255.255, inclusive.)

Example usage

Initialization file:

```
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupport:13100
```

Parameter: groupname

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

groupname

Description

This parameter creates a virtual group that servlet container processes across multiple systems can join for horizontal scaling purposes. Systems that list the same virtual group name are part of the same

group. This parameter requires the use of the *groupmcastaddress* and *groupport* parameters to define the resources group members can use to talk to the load balancer process.

It is best practice to place this parameter in the HP Service Manager initialization file so that all servlet container processes started on the same host share the virtual group name specified by this parameter.

Valid if set from

Server's operating system command prompt

Initialization file (sm.ini)

Requires restart of the Service Manager server?

Yes

Default value

None

Possible values

Any alphanumeric name without spaces

Example usage

Initialization file:

```
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupport:13100
```

Parameter: groupport

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

groupport

Description

This parameter defines the communications port that servlet container processes can use to communicate with the load balancer process in a horizontal scaling implementation. The servlet container processes talk to one another using User Datagram Protocol (UDP) multicasting and must use a common communications port available on all member systems of the virtual group. You must enable

UDP multicasting traffic on your network to use HP Service Manager virtual grouping.

It is best practice to place this parameter in the HP Service Manager initialization file so that all servlet container processes started on the same host share the communications port specified by this parameter.

Valid if set from

Server's operating system command prompt

Initialization file (sm.ini)

Requires restart of the Service Manager server?

Yes

Default value

None

Possible values

Any communications port valid for UDP multicasting

Example usage

Initialization file:

```
groupname:mygroup1
groupmcastaddress:224.0.1.255
groupport:13100
```

Parameter: httpPort

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

httpPort

Description

This parameter defines the communications port that a servlet container process uses to communicate with clients using HTTP. A servlet container process can only have one HTTP port open at a time.

It is best practice to use this parameter from the command line or configuration file for each individual servlet container process you start. This practice makes it easy to identify the HTTP communications port each servlet container uses.

Valid if set from

Server's operating system command prompt

Start up file (sm.cfg or smstart)

Requires restart of the Service Manager server?

No

Default value

None

Possible values

Any valid communications port number

Example usage

Command line: **sm -httpPort:13081**

Parameter: httpsPort

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

httpsPort

Description

This parameter defines the communications port that a servlet container process uses to communicate with clients using HTTPS (SSL-encrypted HTTP). A servlet container process can only have one HTTPS port open at a time. Servlet container processes can only use an HTTPS communications port if the *sslConnector* parameter is enabled. This parameter requires the use of the *sslConnector* parameter.

It is best practice to use this parameter from the command line or configuration file for each individual servlet container process you start. This practice makes it easy to identify the HTTPS communications port each servlet container uses.

Valid if set from

Server's operating system command prompt

Start up file (sm.cfg or smstart)

Requires restart of the Service Manager server?

No

Default value

None

Possible values

Any valid communications port number

Example usage

Command line: **sm -httpsPort:13081**

Parameter: loadBalancer

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

loadBalancer

Description

This parameter creates a load balancer process that listens to incoming client requests on the communications port specified by the httpPort parameter. The load balancer process forwards client connection requests to an available thread on a servlet container process. This parameter requires at least one servlet container process to which to forward client requests. The servlet container process can be on the local HP Service Manager system or on another system that is part of the same virtual group.

Valid if set from

Server's operating system command prompt

Start up file (`sm.cfg` or `smstart`)

Requires restart of the Service Manager server?

No

Default value

None

Possible values

None

Example usage

Command line: **sm -loadBalancer -httpPort:13080**

Parameter: sslConnector

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt.

Parameter

sslConnector

Description

This parameter defines whether servlet container processes have an HTTPS (SSL-encrypted HTTP) communications port available. A servlet container process can only have one HTTPS port open at a time. Servlet container processes can only use an HTTPS communications port if the sslConnector parameter is enabled. This parameter requires the use of the httpsPort parameter.

It is best practice to place this parameter in the HP Service Manager initialization file so that you enable or disable the HTTPS port for all servlet containers on the same system.

Valid if set from

Server's operating system command prompt

Initialization file (sm.ini)

Requires restart of the Service Manager server?

Yes

Default value

1

Possible values

0 (Disable)

1 (Enable)

Example usage

Initialization file: `-sslConnector:0`

Startup options for servlet container processes

The only way you can start the servlet container processes in a servlet implementation is to manually start all needed servlet container processes in advance. HP recommends that you start all needed servlet container processes in advance because it ensures that the host dedicates sufficient system resources to client connections, and because clients do not experience any connection delays while a servlet container process starts up. Each servlet container process requests host system resources when it first started. By starting all processes in advance you ensure that the host has sufficient resources to run your servlet container processes, and you can determine how much spare capacity, if any, the host has. In addition, clients do not have to wait for a servlet container process to start before they can connect. You can manually start servlet container processes requires using the following parameters from the command line or HP Service Manager configuration file (sm.cfg).

Implementation	Command line parameters or configuration file entries
Single servlet implementation	`sm -httpPort:13081 httpsPort:13082`
Horizontal or vertical scaling implementations	`sm -httpPort:13080 -loadBalancer` `sm -httpPort:13081 -httpsPort:13082`

Managing multiple servlet container processes

In vertical and horizontal scaling implementations administrators have the option of managing multiple servlet container processes. Administrators can stop or quiesce individual servlet container processes from the server's operating system command line. This allows you to take servlet container processes offline for maintenance or to manage system resources on the HP Service Manager host.

To quiesce multiple servlet container processes you must use the command line. The quiesce form does not recognize servlet container implementations.

Additionally, you can set a process to monitory the heap memory and take action to prevent memory overload on any one servlet.

Quiesce all servlet container processes in a servlet implementation

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To quiesce all servlet container processes in a servlet implementation:

1. Open the server's operating system command line prompt.
   For example, to open the Windows command prompt:
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
2. Change directories to the RUN folder of your HP Service Manager installation. For example:
   `cd C:\Program Files\HP\ Service Manager\server\RUN`
3. Type the following command:

   ```
   sm -quiesce:2 -group
   ```
4. Press Enter.
   The server quiesces all the servlet container processes in the virtual group.

Quiesce all servlet container processes on a host

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To quiesce all servlet container processes on a host:

1. Open the server's operating system command line prompt.
   For example, to open the Windows command prompt:
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
2. Change directories to the RUN folder of your HP Service Manager installation. For example:
   `cd C:\Program Files\HP\ Service Manager\server\RUN`
3. Type the following command:

   ```
   sm -quiesce:2 -host:<host name>
   ```
4. Press Enter.
   The server quiesces all the servlet container processes on the host.

Shut down all servlet container processes in a servlet implementation

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To shut down all servlet container processes in a servlet implementation:

1. Open the server's operating system command line prompt.
   For example, to open the Windows command prompt:
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
2. Change directories to the RUN folder of your HP Service Manager installation. For example:
   `cd C:\Program Files\HP\ Service Manager\server\RUN`
3. Type the following command:
   `sm -shutdown -group`
4. Press Enter.
   The server shuts down all the servlet container processes in the virtual group.

Shutdown all servlet container processes on a host

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To shut down all servlet container processes on a host:

1. Open the server's operating system command line prompt.
   For example, to open the Windows command prompt:
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
2. Change directories to the RUN folder of your HP Service Manager installation. For example:
   `cd C:\Program Files\HP\ Service Manager\server\RUN`

3. Type the following command:

   ```
   sm -shutdown -host:<host name>
   ```

4. Press Enter.
   The server shuts down all the servlet container processes on the host.

Monitoring memory in Service Manager processes

The Service Manager process embeds a Java Virtual Machine (JVM) so that the heap memory provided by the operating system for any process is partitioned into a Java Heap and a Native Heap. The Service Manager process self–monitors these heaps and takes action to prevent a crash.

During startup of a Service Manager process, immediately after the JVM is created, a memory monitor thread is started, which will poll for a low memory condition every 15 seconds. On every iteration, the memory monitoring thread fetches the current usage of java heap (all platforms) and native heap (Windows only) and set the process global flags `lowMemoryOnJava` and `lowMemoryOnNative` when the usage exceeds 90% of allocated memory in that category. These flags are unset again when the usage of memory drops below 70%.

To prevent server crashes due to low available heap memory and to avoid disrupting users who are logged on to a Service Manager process, the servletcontainer takes the following actions when the low memory flags are set:

- Writes warning messages to `sm.log`.
- Notifies LoadBalancer not to send that servlet any new client connections.
- Denies new client sessions (Windows client, web client or web services client) for that servlet. They are sent to another one.
- Does not allow existing clients to open new tabs. However, they can trade the existing tabs for new ones. For example, if you close two tabs, you will be allowed to open two new tabs. If you cannot operate without opening any new tabs you need to logoff and login again.

Normal processing is resumed when the heap usage drops below 70%. The servletcontainer takes the following actions when the low memory flags are unset.

- Writes Info messages to the sm.log file indicating normal memory usage.
- Notifies LoadBalancer to resume sending of new client connections.

Monitoring native heap and Java heap memory

Java heap

When a Java Virtual Machine (JVM) is started in Service Manager process, the Service Manager process defines the minimum and the maximum amount of memory that is allocated for JVM. The minimum and maximnm vary for different kinds of Service Manager processes. For servlet container processes that needs more java processing, the minimum and maximum defined are 256M. For processes that are shortlived and do not need much java processing, the minimum and maximum defined are 64M and 96M. You can override these default values by using the JVMOption parameters. For example, you could set `JVMOption0:-Xms128M, JVMOption1:-Xmx256M`, to provide minimum and maximum values of 128M and 256M for java heap memory.

The JVM internally divides the allocated heap into 4 different memory pools. The Service Manager process monitor for the low memory condition on "Tenured Gen(Old Generation)" only. Therefore, the low memory condition still can occur on other memory pools like "Eden Space (Young Generation)" or "Survivor Space".

Note: For detailed explanation of Garbage collection and Memory Pools, refer to Sun Microsystem's Memory Management whitepaper".

Native heap

Windows: The Windows operating system has APIs that fetch the current memory usage of a process and the maximum that can be allocated for that process. We monitor the native heap using the API's monitoring thread, and turn the low memory condition processes on or off accordingly.

Unix Systems: Not implemented.

Note: The monitoring the memory of Service Manager processes reduces the chance of a servletcontainer running out of memory, but does not completely prevent it.

Logging memory monitoring

You can turn on detailed messages of the memory monitoring thread by adding the following parameter to the `sm.ini` file:

```
log4jdebug:com.hp.ov.sm.common.oom.LowMemoryHandler
```

Messages like these will be sent to the `sm.log` file when the parameter is turned on. Warning messages are written to the log whether detailed messages are turned on or not.

```
  2120(  2232) 02/04/2009 13:25:02 JRTE I Starting Memory Monitoring thread to
check for memory every 15 seconds.
  2120(  5980) 02/04/2009 13:25:02 JRTE D JavaMemory Max(123928576) Used(1011872)
%Used(0.0)
  2120(  5980) 02/04/2009 13:25:02 JRTE D NativeMemory Max(2147352576) Used
(358121472) %Used(16.0)
  2120(  5980) 02/04/2009 13:28:32 JRTE W Process Low on Java Memory. Max
(123928576) Used(119603800) PercentUsed(96.0)
  2120 02/04/2009 13:30:14 JRTE W Send error response: Server is running low on
memory try again.
  2120 02/04/2009 13:30:16 JRTE W Send error response: Server is running low on
memory try again.
  2120(  5980) 02/04/2009 13:30:47  RTE I Process Java Heap Memory is back to
normal range.
  2120(  5980) 02/04/2009 13:30:47 JRTE I Process Java Memory. Max(123928576)  Used
(4256640) PercentUsed(3.0)
```

Parameter: memorypollinterval

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt. You can always set a startup parameter from the server's operating command prompt.

Parameter

memorypollinterval

Description

This parameter defines the frequency at which the memory heap is monitored. The memory monitoring thread checks for available and maximum memory every *n* seconds, where *n* is the value specified. If you specify zero (0), memory monitoring is disabled.

> **Note:** Setting this parameter too low (for example, every second) places additional load on the system.

Valid if set from

Server's operating system command prompt

Start up file (sm.ini or smstart)

Requires restart of the Service Manager server?

No

Default value

15

Possible values

Number of seconds

Example usage

Command line: **sm -httpPort:13080 -memorypollinterval:15**

Parameter: log4jdebug

Startup parameters change the behavior of the HP Service Manager server. You can always set a startup parameter from the server's operating system command prompt. You can always set a startup parameter from the server's operating system command prompt.

Parameter

log4jdebug:com.hp.ov.sm.common.oom.LowMemoryHandler

Description

This parameter enables certain java packages to be started in debug mode.

Valid if set from

Server's operating system command prompt

Start up file (sm.ini or smstart)

Requires restart of HP Service Manager server

Yes

Default value

None -- By default none of the java packages will be run in debug mode.

Possible values

com.hp.ov.sm.common.oom.LowMemoryHandler

Example usage

Command line: **sm -httpPort:13080 -log4jDebug:com.hp.ov.sm.common.oom.LowMemoryHandler**

Startup and shutdown

There are several methods you can use to start and stop the HP Service Manager server and associated background processes depending upon the operating system on which your system runs. See the topics associated with your Service Manager server's operating system for more information.

Starting Service Manager on UNIX

You must start the HP Service Manager server before users can connect with client sessions. You can also manually start the Service Manager server from the following interface:

- UNIX command line

Start the server from the UNIX command line

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To start the server from the UNIX command line:

1. Change directories to your HP Service Manager Run directory. For example:

   ```
   cd /HP/Service Manager x.xx/Run
   ```

2. Type the following command:

   ```
   smstart
   ```

3. Press ENTER.

Starting Service Manager on Windows

You must start the HP Service Manager server before users can connect with client sessions. By default, the Service Manager server runs as an automatically starting service on Windows. The Service Manager server will start every time that you restart your server. You can also manually start the Service Manager server from the following interfaces:

- Windows services
- Windows command prompt

Configure the HP Service Manager service to run as a Windows user

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To configure the HP Service Manager service to run as a Windows user:

1. Open the Services applet.
   - Click **Start** > **Programs** > **Administrative Tools** > **Services**

     –or–

   - Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**
2. In the services list, click **HP Service Manager**.
3. Stop the service.
   - Click the Stop service button

     –or–

   - Click **Action** > **Stop**
4. Edit the service properties.
   - Click the Properties button

     –or–

   - Click **Action** > **Properties**
5. Click the **Log On** tab.
6. In the **Log on as: This Account** field, click **Browse**.
   Windows displays a list of local domains and user accounts.
7. Select the domain and user account you want to use to run HP Service Manager.

8. Click **OK**.
9. Type the user account password and password confirmation.
10. Click **OK**.
11. Start the service.
    - Click the Start service button
      –or–
    - Click **Action** > **Start**

Start a HP Service Manager client listener

Applies to User Roles:

System Administrator

You can start one or more client listeners to manage your system's Web and Windows client connection requests. The number of client listeners you need to start depends upon the type of servlet implementation you are running.

Note: The client listener cannot accept connection requests from legacy integration products such as SCAuto. For these SCAuto connections, you must start an SCAuto listener (scautolistener).

You must have administrative access to the server operating system to use this procedure.

To start a HP Service Manager client listener:

1. Stop the Service Manager server.
2. Open the Windows command prompt.
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
3. Change directories to the RUN folder of your Service Manager installation. For example:
   `cd C:\Program Files\HP\ Service Manager\server\RUN`
4. Type the one of the following commands:

Servlet implementation option	Command line parameters or configuration file entries
Single servlet implementation	`sm -httpPort:13081 httpsPort:13082`
Horizontal or vertical scaling implementations	`sm -httpPort:13080 -loadbalancer` `sm -httpPort:13081 -httpsPort:13082`

Each client listener listens to client connection requests on a separate communications port. You can specify your own communications port for the example port above.

Start the server from the Windows command prompt

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To start the server from the Windows command prompt:

1. Open the Windows command prompt.
   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.
2. Change directories to the RUN folder of your HP Service Manager installation. For example:

   ```
   cd C:\Program Files\HP\ Service Manager x.xx\Server\RUN
   ```
3. Type the following command:

   ```
   sm -httpPort:13080 -httpsPort:13081
   ```

   Tip: You can omit the **httpPort** and **httpsPort** parameters if you provide them in the Service Manager initialization file (sm.ini).
4. Press Enter.

   Caution: You must leave the command prompt window open while the Service Manager server runs. Closing the command prompt window stops Service Manager immediately without cleaning up any processes or releasing any record locks.

Start the server from Windows services

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To start the server from Windows services:

1. Open the Services applet.
   - Click **Start** > **Programs** > **Administrative Tools** > **Services**
     –or–
   - Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**
2. In the services list, click **HP Service Manager**.
3. Start the service.
   - Click the Start Service link
     –or–
   - Click **Action** > **Start**

   Windows displays a message that the Service Manager service is starting. After several seconds, the service starts and displays Started in the Status field. If the Service Manager service does not start, contact customer support with any error messages.

Stopping Service Manager on UNIX

Stopping the HP Service Manager server immediately disconnects all client sessions and ends all currently running Service Manager processes. By default, the Service Manager server runs continuously and has no scheduled shutdown times.

There are two methods for stopping Service Manager:

- Server shutdown with a warning message — allows the administrator to broadcast a warning message before stopping the server
- Server stop from outside a client — allows the administrator to stop the server without first connecting to the server through a client

Server shutdown with a warning message

When Service Manager is running on a UNIX server, you can issue a shutdown warning from the following interfaces:

- From a Service Manager client when you are logged in as a user with system administrator capabilities.
- From the UNIX command line when you use the **smstop** script

Either shutdown option allows you to broadcast a warning message to currently connected users letting them know that you are planning to shut the server down. However, only the shutdown option from the client also allows you to specify in advance when you want to shut a server down.

There are three types of shutdown available to administrators from the client:

- Immediate — stops the server as soon as the administrator confirms the shutdown request. The server issues no warning messages before stopping.
- Delayed — stops the server after a delay specified by the administrator. The server issues a warning message every minute during the warning interval. The warning interval determines how many minutes before stopping the server that the server starts issuing warning messages.
- Scheduled — stops the server at a specified date and time. The server issues a warning message every minute during the warning interval. The warning interval determines how many minutes before stopping the server that the server starts issuing warning messages.

The shutdown option from the UNIX command line does not allow you to delay or schedule a server shutdown.

Server stop from outside a client

You must have administrative access to the server operating system to stop the Service Manager server outside of a client. If you want to issue a warning message before stopping the server, you must use the **smstop** script. Unless you know that there are no active client connections, it is generally preferable to shutdown a server with a warning message rather than stop it.

You can stop a Service Manager server running on UNIX from the following interface without using a client connection:

- UNIX command line

Stop the server from the UNIX command line

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To stop the server from the UNIX command line:

Note: This procedure stops the server without displaying any warnings to currently connected users. If you want to broadcast a warning message before stopping the server, use one of the shutdown options available from the HP Service Manager clients.

1. Change directories to your Service Manager Run directory. For example:

   ```
   cd /HP/Service Manager x.xx/Run
   ```

2. Type one of the following commands:

Command	Description
sm -shutdown	Immediately stops the Service Manager server and all related processes.
sm -shutdown -group	Used in a horizontally-scaled environment. Immediately stops the Service Manager server and all related processes, including Service Manager processes on remote hosts in the same group.
smstop	The script attempts to do a normal shutdown of the HP Service Manager system and, if successful, stops any remaining Service Manager processes and cleans up IPC resources. **Note**: The `scdb.system` record in the `info` table acts as a Service Manager database lock, preventing multiple Service Manager groups from connecting to the same database. This record will only be removed during normal shutdown and when there are no more Service Manager processes connecting to that database. In a vertically-scaled environment, normal shutdown during `sm -shutdown`, `smstop`, or `smstop -f` removes this record. In a horizontally-scaled environment, only a normal shutdown (`sm -shutdown -group`) shuts down the whole group.
smstop -d	The script runs the shutdown script in debug mode. The system produces verbose output messages in debug mode.
smstop -f	The script forces a shutdown of the Service Manager system if the normal shutdown process fails. By default, the script will wait for up to ten minutes for the normal shutdown process to complete. To change the wait time, you can set the SMSTOP_WAIT environment variable with the number of seconds you want the script to wait for the completion of the normal shutdown process (the default value is 600 seconds). For example, to change the wait time to 15 minutes, you can use the following command: `SMSTOP_WAIT=900 smstop -f`

Command	Description
smstop -F	The script immediately forces a shutdown of the Service Manager system, bypassing the normal shutdown process. **Caution:** Forcing a system shutdown can result in file corruption if there is file regeneration in progress at the time of the shutdown. HP recommends using the Service Manager status and system.monitor functions to determine if there are any active processes running prior to forcing a system shutdown.
smstop -h	Displays help information for the smstop script.
smstop -q	Runs the shutdown script in quiet mode. The system does not produce output messages in quiet mode.

3. Press ENTER.

Stopping Service Manager on Windows

Stopping the HP Service Manager server immediately disconnects all client sessions and ends all currently running Service Manager processes. By default, the Service Manager server runs continuously and has no scheduled shutdown times.

There are two methods for stopping Service Manager:

- Server shutdown with a warning message — allows the administrator to broadcast a warning message before stopping the server
- Server stop from outside a client — allows the administrator to stop the server without first connecting to the server through a client

Server shutdown with a warning message

When Service Manager is running on a Windows server, you can only issue a shutdown warning from a Service Manager client, and then only if you are logged in as a user with system administrator capabilities. The shutdown option allows you to broadcast a warning message to currently connected users letting them know that you are planning to shut the server down. The shutdown option also allows you to specify in advance when you want to shut a server down.

There are three types of shutdown available to administrators:

- Immediate — stops the server as soon as the administrator confirms the shutdown request. The server issues no warning messages before stopping.
- Delayed — stops the server after a delay specified by the administrator. The server issues a warning message every minute during the warning interval. The warning interval determines how many minutes before stopping the server that the server starts issuing warning messages.
- Scheduled — stops the server at a specified date and time. The server issues a warning message every minute during the warning interval. The warning interval determines how many minutes before stopping the server that the server starts issuing warning messages.

Server stop from outside a client

You must have administrative access to the server operating system to stop the Service Manager server outside of a client. When Service Manager is running on a Windows server, you cannot issue a warning message if you stop the server outside of a client. Unless you know that there are no active client connections, it is generally preferable to shutdown a server with a warning message rather than stop it.

You can stop a Service Manager server running on Windows from the following interfaces without using a client connection:

- Windows services
- Windows command prompt

Stop the server from the Windows command prompt

Applies to User Roles:

System Administrator

You must have administrative access to the server operating system to use this procedure.

To stop the server from the Windows command prompt:

Warning: This procedure stops the server without displaying any warnings to currently connected users. If you want to broadcast a warning message before stopping the server, use one of the shutdown options available from the HP Service Manager clients.

1. Open another Windows command prompt.

   Click **Start** > **Programs** > **Accessories** > **Command Prompt**.

   Caution: Do not close the command prompt window running Service Manager. Closing the command prompt window stops Service Manager immediately without cleaning up any processes or releasing any record locks.

2. Change directories to the RUN folder of your Service Manager installation. For example:

   ```
   cd C:\Program Files\HP\ Service Manager x.xx\Server\RUN
   ```

3. Type the following command:

   ```
   sm -shutdown:0
   ```

4. Press ENTER.

Stop the server from Windows services

You must have administrative access to the server operating system to use this procedure.

Note: This procedure stops the server without displaying any warnings to currently connected users. If you want to broadcast a warning message before stopping the server, use one of the shutdown options available from the HP Service Manager clients.

1. Open the services applet.
   - Click **Start** > **Programs** > **Administrative Tools** > **Services**

     –or–

   - Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**

2. In the services list, click **HP Service Manager**.

3. Stop the service.
   - Click the Stop Service button

     –or–

   - Click **Action** > **Stop**

   Windows displays a message that the Service Manager service is stopping. After several seconds, the service stops and displays a blank entry in the Status field. If you see any error messages, contact customer support with the message.

Server shutdown

A system administrator can shutdown a HP Service Manager server from any client. Shutting down a server differs from manually stopping a server in the following ways:

- You can schedule a shutdown in advance
- The server can broadcast warning messages to all connected users before the server stops
- The server stops all active processes before shutting down

In a production environment, it is generally preferable to shutdown a server rather than to stop it to ensure data integrity and to allow users to log off.

Do an immediate shutdown of the server

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

Note: This procedure stops the server without displaying any warnings to currently connected users. If you want to broadcast a warning message before stopping the server, use the broadcast form (system.status.broadcast) before you shut down the server.

To do an immediate shutdown of the server:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Shutdown**.
   The System Shutdown Confirmation form opens.
2. Click **Shutdown Now**.
3. Click **Confirm**.
   The HP Service Manager server stops disconnecting all currently connected clients and refusing any further client connections until the server has been restarted.

   Note: Service Manager also disconnects the client you used to shut down the server.

Do a delayed shutdown of the server

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

To do a delayed shutdown of the server:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Shutdown**.
   The System Shutdown Confirmation form opens.
2. Click **Advise and Shutdown**.
3. In the text field provided, type the number of seconds you want the system to wait before it shuts down.
4. Click **Advise and Shutdown**.

   HP Service Manager broadcasts the following message to all connected users:

   System will SHUTDOWN in <n> seconds; please finish and log off.

   The value *<n>* is the number of seconds you entered for the delay. After the specified number of seconds passes, HP Service Manager stops disconnecting all currently connected clients and refusing any further client connections until the server has been restarted.

   Note: Service Manager also disconnects the client you used to shutdown the server.

Do a scheduled shutdown of the server

Applies to User Roles:

System Administrator

You must have the **SysAdmin** capability word to use this procedure.

To do a scheduled shutdown of the server:

1. Click **System Administration** > **Ongoing Maintenance** > **System** > **Schedule Shutdown**.
   The Automatic Shutdown Information form opens.

2. Click **Search**.

   HP Service Manager displays a list of existing shutdown records.

   Note: HP Service Manager can only have one active shutdown record at a time. If a shutdown record already exists, update the existing record rather than creating a new one.

3. Type or select the following information.
   - **Shutdown Date** — type the date and time you want the server to shutdown
   - **Auto-Shutdown** — select this option if you want the shutdown to be automatic
   - **Reschedule Interval** — type or select the number of days you want the server to wait for the next scheduled shutdown. If you leave this entry blank, then HP Service Manager uses the default value of one day.
   - **Shutdown Warning** — type or select the number of minutes you want the server to issue shutdown warnings, if any

4. Do one of the following:
   - If there is no existing shutdown record, click **Add**.
     HP Service Manager displays the following message:
     shutdown record added.
   - If there is an existing shutdown record, click **Save**.
     HP Service Manager displays the following message:
     shutdown record updated.

5. From the pull-down options menu, click **Update schd**.
   HP Service Manager displays the following message:
   shutdown record updated.
   At the specified date and time, HP Service Manager stops disconnecting all currently connected clients and refusing any further client connections until the server has been restarted.

System information record

The system information record contains common information that all HP Service Manager applications can use.

From the company record administrators can do the following.

Note: The information listed below is presented in the same order within the tabs displayed in this form.

- Add company contact information
- Add Web tier and self-service URL information
- Set general login restrictions
- Set user lockout conditions
- Define the operator template to be applied to LDAP users
- Set user account expiration conditions
- Enable other applications to integrate with HP Service Manager
- Define default menu information
- Set password reset restrictions
- Enable password history
- Set password format restrictions
- Set password lifetime conditions
- Set the time zone the system uses
- Set the date format the system uses
- Enable legacy print routines
- Define the months of the year and the abbreviations the system uses

- Enable or select miscellaneous settings
  - Set the default language the system uses
  - Set the default currency the system uses
  - Set the maximum size for a file attachment
  - Set the maximum number of inbox views that a user can create
  - Set the maximum memory all file attachments can use
  - Enable case sensitivity
  - View the system start time
  - Enable message queues
  - Enable auditing in the system log
  - Enable multi-company mode
  - Enable learning for HP Service Manager Knowledge Bases
  - Enable editing of the learning record
  - Enable universal search
  - Disable the pop-up window to reload records
  - Enable Calendar and Time Period Management
  - Enable Enhanced Query Hash
- List message processors the system uses

Add company contact information

Applies to User Roles:

System Administrator

To add company contact information:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Type the company contact information.
3. Click **Save**.

Add the Web tier and self-service URLs

Applies to User Roles:

System Administrator

You can add the Web tier and self-service URLs to the system information record to make them available as system variables.

To add the Web tier and self-service URLs:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Active Integrations** tab.
3. Type values for the **Web Server URL** and **ESS URL** fields.
4. Click **Save**.

Enable the enhanced query hash algorithm

Applies to User Roles:

System Administrator

Starting with version 9.41, Service Manager provides an enhanced query hash algorithm, which ensures a higher level of system security when the Service Manager web client or Service Request Catalog (SRC) is handling URLs. By default, this functionality is disabled for backward compatibility.

To enable the enhanced query hash algorithm, follow these steps:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **General** tab.

3. Configure the enhanced query hash functionality as described in the following table.

Option	Default value	Description
Enable Enhanced Query Hash	false	When this option is set to true (selected), the Service Manager server generates query hashes based on the enhanced query hash algorithm. For security considerations, we recommend you to enable this option. By default, this option is disabled so that your system can continue to use the legacy algorithm and work as before.
Allow Legacy Query Hash	true	This option is available only when the **Enable Enhanced Query Hash** option is enabled. By default, this option is enabled so that the Service Manager web client or SRC can accept URLs with query hashes that are generated based on either the legacy or the enhanced algorithm. If you deselect this option, the client displays an error message when processing URLs with a legacy query hash, indicating the hash code is invalid. **Note:** Use this option only during your transition from the legacy algorithm to the enhanced algorithm. Once no URLs with legacy query hashes exist in your system, we recommend you to disable this option.

4. Click **Save**.

Set the default maximum number of login attempts per session

Applies to User Roles:

System Administrator

To set the default maximum number of login attempts per session:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.

2. Click the **Logon Info** tab.

3. In the **Attempts Per Login Session** field, type the number of login attempts to use as the company default.
4. Click **Save**.

The Windows client creates a new session for each login attempt. There will never be more than one login attempt per session. To restrict login attempts from the Windows client, enable the **Use User Lockout** option and set a value for the **Attempts Until Lockout** field. Use the **Attempts Per Login Session** setting to control Web client login attempts.

Set the default maximum number of sessions per user

Applies to User Roles:

System Administrator

To set the default maximum number of sessions per user:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Logon Info** tab.
3. In the **Max Logins Per User** field, type the number of user sessions that are the company default.
4. Click **Save**.

Set the default password reset

Applies to User Roles:

System Administrator

To set the default password reset:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Passwords** tab.

3. Click the **Password Standards** tab.
4. Select one of the following options:
   - **Prevent Pwd Reset** — select this option to disable the password reset option.
   - **Reset to User Name** — select this option to reset passwords to the operator's login name.
   - **Prompt for Value** — select this option to have HP Service Manager prompt the system administrator for a new password.

     Tip: For increased security, the system administrator can also select the password expired option in the operator record. Doing so requires reset users to change their passwords at the next login.
   - **Reset to Value** — select this option to type a global password that applies to all reset users.
5. Click **Save**.

Set the default user expiration interval

Applies to User Roles:

System Administrator

To set the default user expiration interval:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Logon Info** tab.
3. In the **Expiration Interval** field, type the time period that a user can remain inactive until the user account expires.
   Type the time period in the following format: *Dayshours*:*minutes*:*seconds*. For example, 4 03:02:01 expires the user account after 4 days, 3 hours, 2 minutes, and 1 second of inactivity.
   Important: The system administrator cannot reset an expired account. The system administrator must create a new operator record for the expired account.
4. Click **Save**.

Set the default user inactivation interval

Applies to User Roles:

System Administrator

To set the default user inactivation interval:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Logon Info** tab.
3. In the **Inactivation Interval** field, type the time period that a user can remain inactive until HP Service Manager locks out the account.
   Type the time period in the following format: *Dayshours*:*minutes*:*seconds*. For example, 4 03:02:01 locks out the user after 4 days, 3 hours, 2 minutes, and 1 second of inactivity.
4. Click **Save**.

Setting file attachment limits

Setting attachment size limits

You can set the following two file attachment size limits:

- Maximum single attachment size

  This value limits the maximum size of a single attachment.

- Total attachment size

  This value limits the total size of all attachments in an attachment container, such as an incident record.

You can set these attachment size limits in the following ways:

- Specify the attachment size limit for the entire system in the system information record
- Specify the attachment size limit for a particular operator in their operator record

- Specify the attachment size limit for the attachment container on a form by using Forms Designer
- Specify the attachment size limit for individual attachments by using the *MaxAttachUploadSize* and *MaxTotalAttachUploadSize* parameters.

  Note: These parameters are located in the web.xml file. The default value of the *MaxAttachUploadSize* parameter is 10 MB, and the default value of the *MaxTotalAttachUploadSize* parameter is 50 MB.

If you do not specify a value in one of these places, there is no limit on the size of attachments, and users can add attachments as large as the database record will hold. If you specify the attachment size limits in more than one place, then the order of precedence is:

1. The value from the attachment container
2. The value from the operator record
3. The value from the company record
4. The values of the *MaxAttachUploadSize* and *MaxTotalAttachUploadSize* parameters

As an example, for the maximum size of a single attachment, if the company record limit is set to 1,000,000 bytes, the operator has a limit set to 2,000,000 bytes, and the container has a limit of 3,000,000 bytes, the person will be able to store a 3,000,000 byte attachment in the container (the attachment container property overrides the other two limits). A different operator without a limit, storing attachments in a different container with no limit set, would only be able to save a 1,000,000 byte attachment.

Note: File size validation is not performed at the widget level on the web service. Therefore, it is possible to attach a file that exceeds the configured size limits by using the web service.

Setting attachment count limits

You can limit the number of individual files that can be attached to each record in Service Manager. To do this, configure the *maxattachmentcount* parameter. By default, the value of this parameter is 100 files.

Set the maximum file attachment size for the entire company

Applies to User Roles:

System Administrator

To set the maximum file size of a single attachment for the entire company:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **General** tab.
3. In the **Max Attachment Size** field, type the number of bytes for the maximum file attachment size.
4. Click **Save**.

You can define a file attachment maximum size limit for all users from the system wide company record. The system wide company record file attachment size limit is superseded by size limits defined in individual operator records and for attachment containers defined in Forms Designer.

Set the total file attachment size for the entire company

Applies to User Roles:

System Administrator

To set the total attachment size in any attachment container for the entire company:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **General** tab.
3. In the **Total Attachment Size** field, type the number of bytes for the total file attachment size.
4. Click **Save**.

You can define a total attachment size limit for all users from the system wide company record. The total attachment size limit defined in system wide company record file is superseded by size limits defined in individual operator records and for attachment containers defined in Forms Designer.

Set the maximum number of attachments for each record

Delete this text and replace it with your own content.

Set the maximum number of inbox views

Applies to User Roles:

System Administrator

To set the maximum number of inbox views that a user can create:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Select the **General** tab.
3. In the **Max Views Shown in the List** field, type the maximum number of inbox views that users are allowed to create.

   Notes:

   - The default number of allowable views is 100.
   - There is no limit to the number of views that can be set.
4. Click **Save**.

Set the menu prompt

Applies to User Roles:

System Administrator

To set the menu promptl:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Menu Information** tab.
3. In the **Menu Prompt** field, type the text or variable to display in the first workspace tab.
4. Click **Save**.

The menu prompt is the text HP Service Manager displays in the first workspace tab.

Activate the command/search line toggle button

Applies to User Roles:

System Administrator

To activate the command/search line toggle button:

1. Follow the steps in Enable an operator to see the command line.
2. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
3. Click the **General** tab.
4. Mark the **Enable Universal Search** check box.
5. Click **Save**.
6. Restart your web applications server.
7. Click **System Status** and verify the lister background process is running. If not, click **Start Scheduler** and double-click **lister.startup** in the list to launch this process.
8. Log out of Service Manager and then log in again. The command/search line toggle button is enabled for the operator.

Time zones

A time zone is Greenwich Mean Time (GMT) plus or minus the required offset for geographic location. HP Service Manager uses time zone information to display local time correctly, and calculate the correct escalation time for alerts. HP Service Manager uses time zone information for:

- Work schedules
- Alerts
- Request Management lead time calculations
- Service Level Agreements (SLAs) and service contracts
- Clocks
- Reminders

System administrators can define a system-wide default time zone and date format in the company record. Individual operator records can contain individual time zone values that override the system-wide default time zone.

HP Service Manager contains out-of-box time zone records for most regions of the world. System administrators can also create their own time zone records. Each time zone record contains:

- Local GMT offset
- Local time switch over definitions
- Local date format

Associated tables

HP Service Manager uses information in these tables to track time zone information.

- tzfile
- contact
- location

- device
- assignment
- cm3groups
- ocmgroups

Set the default system time zone

Applies to User Roles:

System Administrator

To set the default system time zone:

1. Click **System Administration** > **Base System Configuration** > **Miscellaneous** > **System Information Record**.
2. Click the **Date Info** tab.
3. In the **Time Zone** field, type or select the default system time zone.
4. Click **Save**.
5. Click **More** or the More Actions icon, and then select **Set Parms**.
6. Stop and restart the HP Service Manager server.

Add a time zone record

Applies to User Roles:

System Administrator

To add a time zone record:

1. Click **Tailoring** > **Database Manager**.
2. In the **Form** field, type `tzfile`.
3. Click **Search**.

4. Specify information for the following fields:
   - **Time Zone Name** identifies the time zone.
   - **Description** specifies the locations where the time zone applies.
   - **Default Date Format** is one of these date formats:
     - **mm/dd/yy** — standard U.S. date format
     - **dd/mm/yy** — standard European date format
     - **yy/mm/dd** — optional date format
     - **mm/dd/yyyy** — standard U.S. date format
     - **dd/mm/yyyy** — standard European date format
     - **yyyy/mm/dd** — optional date format

     The value you enter in this field overrides the default date format in the company record.
   - **Date Separator** is the character that separates the year, month, and date. Only "/","-", or "." is allowed.
   - **Time Change Definitions** defines every possible time offset that can occur within the time zone:
     - **Name (abbr)** is the abbreviation for a particular time offset.
     - **Offset from GMT** is the number of hours, minutes, and seconds the time zone is ahead or behind Greenwich Mean Time (GMT) in this format:
       plus or minushours:*minutes*:*seconds*.
       For example, +04:03:02 is ahead of Greenwich Mean Time by 4 hours, 3 minutes, and 2 seconds.
   - **Time Change Dates/Indexes** defines when time changes occur:
     - **Switchover Date** is the date and time when the offset begins.
       Type the date and time in this format: *datehour*:*minute*:*second*.
       For example, 04/04/04 02:00:00 switches over on April 4, 2004 at 2 AM.
     - **Index of Definition** is the index number of the time offset used on the switch over date.
       The index number matches the position the offset has in the Time Change Definitions. The first offset is index 0, the second offset is index 1, and so on.
5. Click **Add**.

Your user profile must include the system administration capability word to add this record.

Delete a time zone record

Applies to User Roles:

System Administrator

To add a time zone record:

1. Click **Tailoring** > **Database Manager**
2. In the **Form** field, type **tzfile**.
3. Click **Search**
4. Type or select optional search criteria.
5. Click **Search**.
6. Select the time zone to delete.
7. Click **Delete**.
8. Click **Yes** to confirm the deletion.

Update a time zone record

Applies to User Roles:

System Administrator

To update a time zone record:

1. Click **Tailoring** > **Database Manager**.
2. In the **Form** field, type `tzfile`.
3. Click **Search**.
4. Update one or more of these fields:
   - **Time Zone Name** identifies the time zone.
   - **Description** specifies the locations where the time zone applies.

- **Default Date Format** is one of these date formats:
  - **mm/dd/yy** — standard U.S. date format
  - **dd/mm/yy** — standard European date format
  - **yy/mm/dd** — optional date format
  - **mm/dd/yyyy** — standard U.S. date format
  - **dd/mm/yyyy** — standard European date format
  - **yyyy/mm/dd** — optional date format

  The value you enter in this field overrides the default date format in the company record.
- **Time Change Definitions** defines every possible time offset that can occur within the time zone:
  - **Name (abbr)** is the abbreviation for a particular time offset.
  - **Offset from GMT** is the number of hours, minutes, and seconds the time zone is ahead or behind Greenwich Mean Time (GMT) in this format:
    plus or minushours:*minutes*:*seconds*.
    For example, +04:03:02 is ahead of Greenwich Mean Time by 4 hours, 3 minutes, and 2 seconds.
- **Time Change Dates/Indexes** defines when time changes occur:
  - **Switchover Date** is the date and time when the offset begins.
    Type the date and time in this format: *datehour*:*minute*:*second*.
    For example, 04/04/04 02:00:00 switches over on April 4, 2004–2005 at 2 AM.
  - **Index of Definition** is the index number of the time offset to use when it reaches the switch-over date.
    The index number matches the position the offset has in the Time Change Definitions. The first offset is index 0, the second offset is index 1, and so on.

5. Click **Save**.

View a time zone record

Applies to User Roles:

System Administrator

To view a time zone record:

1. Click **Tailoring** > **Database Manager**.
2. In the **Form** field, type `tzfile`.
3. Click **Search**.
4. Type or select optional search criteria.
5. Click **Search**.
6. Select the time zone to view.

UTF-8 conversion

The HP Service Manager server can convert system files to UTF-8 format on-demand or in the background. This allows an existing HP Service Manager implementation to remain available during the upgrade process. You can either let the server convert files as they are requested or start a background process to convert files. See the HP Service Manager Upgrade Guide for instructions.

UTF-8 (Unicode) support

UTF-8 is part of the Unicode standard, which enables you to encode text in practically any script and language. HP Service Manager supports UTF-8 as the encoding method for new or existing data. It can support multiple languages that adhere to the Unicode standard on the same server.

Service Manager can also convert data to UTF-8 on-demand, thus eliminating system outages.

Prior to Unicode, languages were grouped into sets often referred to as Latin 1 (Western European language such as English, French, German), Latin 2 (Eastern European languages such as Czech, Polish, Slovak), Latin 5 (Turkish) and so on. Earlier HP ServiceCenter versions supported multiple languages only within a language group. Therefore, a single server instance could support French and German, English and Turkish, English and Japanese, but not Turkish and Czech, or Polish and Japanese. The ability to display data in disparate languages from a single server is attractive to any Service Manager user with an international customer base.

Consider the following points to ensure successful Unicode support:

- Start a Service Manager server with the `language:utf8` parameter embedded in the **sm.ini** file or from the command line. This instructs the server to move data in UTF-8 format to external sources, such as exporting data to a text file. This is a transparent conversion of existing data on an as-needed basis.
- Ensure that your RDBMS is correctly configured for UTF-8 support. For more information, see your local Database Administrator.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on System Installation and Setup help topics for printing (Service Manager 9.41)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-ITSM@hp.com.

We appreciate your feedback!


Java™
COMPATIBLE