

HP SOA Systinet

Software Version: 3.20

Installation and Deployment Guide

Document Release Date: July 2009
Software Release Date: July 2009



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2003-2009 Hewlett-Packard Development Company, L.P.

Contents

About this Guide.	5
In this Guide.	5
Document Conventions.	6
Documentation Updates.	6
Support.	7
1 Getting Started.	9
Prerequisites.	9
Supported Platforms.	10
Designing Your Deployment.	15
2 Basic Installation.	19
Using the GUI Installer.	19
GUI Installation Options.	26
After GUI Installation.	34
3 Setting Up Production Environments.	37
SOA Systinet Self-Test.	37
Preparing Databases.	40
Setting Up Application Servers.	52
LDAP Accounts Integration.	118
Siteminder Integration.	120
4 Deploying SOA Systinet.	123
Installation Command Line Options.	124
Unpacking the Distribution.	126
Configuring the Deployment.	126
Preparing Extensions.	139
Preparing Updates.	140

Decoupled Database Deployment.	140
Finishing Installation.	142
Deploying the EAR File.	143
Deploying to Environments without a JDK.	154
5 After Installation.	157
Launching SOA Systinet on JBoss.	157
Enabling SSO in JBoss.	158
Setting up the User Store in JBoss.	158
Logging.	161
Enabling Full Text Search.	168
SSL Certificates.	176
Configuring LDAP over SSL/TLS.	183
Configuring for Non-Latin HTTP Parameters.	184

About this Guide

Welcome to HP SOA Systinet, the foundation of Service Oriented Architecture, providing an enterprise with a single place to organize, understand, and manage information in its SOA. The standards-based architecture of SOA Systinet maximizes interoperability with other SOA products.



HP Software controls access to components of SOA Systinet with a license. This document describes the full functionality of SOA Systinet including licensed components. If your license does not include these licensed components, their features are not available.

In this Guide

This guide describes how to set up an environment and deploy SOA Systinet to it.

This guide contains the following chapters:

- [Chapter 1, Getting Started](#)

Design your environment for SOA Systinet.

- [Chapter 2, Basic Installation](#)

Use the GUI Installer, designed for test and evaluation installations of SOA Systinet.

- [Chapter 3, Setting Up Production Environments](#)

Set up and configure a production environment for SOA Systinet.

- [Chapter 4, Deploying SOA Systinet](#)

Deploy SOA Systinet to a production environment using command-line installation.

- [Chapter 5, After Installation](#)

Perform additional processes that may be required after deploying SOA Systinet.

Document Conventions

This document uses the following typographical conventions:

run.bat make	Script name or other executable command plus mandatory arguments.
<code>--help</code>	Command-line option.
either or	Choice of arguments.
<i>replace_value</i>	Command-line argument that should be replaced with an actual value.
<code>{arg1 arg2}</code>	Choice between two command-line arguments where one or the other is mandatory.
<code>java -jar hpsystinet.jar</code>	User input.
<code>C:\System.ini</code>	File names, directory names, paths, and package names.
<code>a.append(b);</code>	Program source code.
<code>server.Version</code>	Inline Java class name.
<code>getVersion()</code>	Inline Java method name.
Shift+N	Combination of keystrokes.
Service View	Label, word, or phrase in a GUI window, often clickable.
OK	Button in a user interface.
New→Service	Menu option.

Documentation Updates

This guide's title page 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:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport logon page.

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

Support

You can visit the HP Software Support Web site at:

<http://www.hp.com/go/hpsoftwaresupport>

HP Software Support Online 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 HP Software Support web site to:

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

1 Getting Started

Before installing SOA Systinet you must make sure that the environment you want to install to is appropriate and suitable for your needs.

The following sections describe the requirements and options available:

- [Prerequisites on page 9](#)
- [Supported Platforms on page 10](#)
- [Designing Your Deployment on page 15](#)

Prerequisites

This section lists the hardware and software requirements required to run SOA Systinet.

Hardware

Hardware requirements vary depending on sizing and deployment type.

Distributed production environment require the following hardware:

- For each physical node, an Intel Pentium Dual Core processor, 2 GB RAM, 1 GB free disk space, and a network card that supports 1 Gb/sec.
- Network bandwidth of 1 Gb/sec or higher.

For development and evaluation purposes, SOA Systinet can run on a single machine, even on a notebook.

The hardware requirements in this case are:

- Intel Pentium IV processor, 1 GB RAM, 1–2 GB free disk space and a network card that supports 100 Mb/sec.

- Network bandwidth of 100Mb/sec or higher.

Software

Each physical node must have the following software:

- A *Java SE Development Kit (JDK)* and a *Java 2 Platform Enterprise Edition (J2EE)* application server from the list in [Supported Platforms on page 10](#). The application server must use this JDK.
- A `JAVA_HOME` environment variable set to point to the Java JDK used by the host J2EE application server.



If you have both a JDK and JRE installed, `JAVA_HOME` must point to the JDK.

- Access to a supported database from [Supported Platforms on page 10](#).

Supported Platforms

SOA Systinet supports the following browsers:

- Microsoft Internet Explorer 6 and 7
- Mozilla Firefox 3

SOA Systinet supports the following mail clients:

- Microsoft Outlook 2003 and 2007
- Mozilla Thunderbird 2
- GMail

SOA Systinet supports the following LDAP implementations:

- Sun ONE Directory Server 5.2
- Microsoft Windows Server 2008 Active Directory

SOA Systinet requires Adobe Flash Player version 9.0 or newer.

SOA Systinet supports the following combinations of application servers, JDKs, and backend databases.

The combinations are described in the following tables, organized by operating system:

- [Supported Platforms on Windows on page 11](#)
- [Supported Platforms on Solaris on page 13](#)
- [Supported Platforms on Linux on page 12](#)
- [Supported Platforms on AIX on page 13](#)
- [Supported Platforms on HP-UX on page 13](#)

The supported drivers for each supported database are listed in [Supported Databases on page 14](#).

[Supported Product Integrations on page 14](#) lists the products that SOA Systinet integrates with.

Supported Platforms on Windows

Table 1, "[Windows Server 2003 on x86 Platforms](#)" lists the application servers, JDKs, and backend databases supported by SOA Systinet on Windows Server 2003 on x86 platforms.

Table 1. Windows Server 2003 on x86 Platforms

J2EE Application Server	Java JDK	Relational Database
BEA® WebLogic Server ® 9.2 MP3	Sun JDK 5.0 update 9	Oracle 10.2.0.4
		MSSQL 2005 (SP2)
IBM WebSphere® 6.1.0.23	built-in JDK	Oracle 10.2.0.4
		DB2 9.1.5
JBoss® 4.2.2	Sun JDK 5.0 update 9	Oracle 10.2.0.4
JBoss® EAP 4.3.0	Sun JDK 5.0 update 9	Oracle 10.2.0.4
		MSSQL 2005 (SP2)

Table 2, "Windows Server 2008 on x86 Platforms" lists the application servers, JDKs, and backend databases supported by SOA Systinet on Windows Server 2008 on x86 platforms.

Table 2. Windows Server 2008 on x86 Platforms

J2EE Application Server	Java JDK	Relational Database
Oracle WebLogic Server 10g Release 3 (10.3)	Sun Java 2 JDK 6.0+	Oracle 10.2.0.4
		MSSQL 2005 (SP2)
IBM WebSphere® 6.1.0.23	built-in JDK	Oracle 10.2.0.4



If you use JBoss, install it with a path that contains less than 20 characters. This limitation is caused by JBoss expanding the application in the local disk and the Windows 255 character limit on path names.

Supported Platforms on Linux

Table 3, "RedHat Enterprise Linux 5.0 on x86 Platforms" lists the application servers, JDKs, and backend databases supported by SOA Systinet on RedHat Enterprise Linux 5.0 on x86 platforms.

Table 3. RedHat Enterprise Linux 5.0 on x86 Platforms

J2EE Application Server	Java JDK	Relational Database
Oracle WebLogic Server 10g Release 3 (10.3)	Sun Java 2 JDK 6.0+	Oracle 10.2.0.4
IBM WebSphere® 6.1.0.23	built-in JDK	Oracle 10.2.0.4
JBoss® EAP 4.3.0	Sun JDK 5.0 update 9	Oracle 10.2.0.4

Table 4, "SLES 10 64-bit on 64-bit x86 Platforms" lists the application servers, JDKs, and backend databases supported by SOA Systinet on SLES 10 64-bit on 64-bit x86 platforms.

Table 4. SLES 10 64-bit on 64-bit x86 Platforms

J2EE Application Server	Java JDK	Relational Database
Oracle WebLogic Server 10g Release 3 (10.3)	Sun Java 2 JDK 6.0+	Oracle 10.2.0.4

J2EE Application Server	Java JDK	Relational Database
IBM WebSphere® 6.1.0.23	built-in JDK	Oracle 10.2.0.4

Supported Platforms on Solaris

Table 5, "Solaris 10 on Sparc" lists the application servers, JDKs, and backend databases supported by SOA Systinet on Solaris 10 on Sparc platforms.

Table 5. Solaris 10 on Sparc

J2EE Application Server	Java JDK	Relational Database
BEA® WebLogic Server ® 9.2 MP3	Sun JDK 5.0 update 9	Oracle 10.2.0.4
IBM WebSphere® 6.1.0.23	built-in JDK	DB2 9.1.5
JBoss® EAP 4.3.0	Sun JDK 5.0 update 9	Oracle 10.2.0.4
Oracle Application Server 10.1.3.4	Sun JDK 5.0 update 9	Oracle 10.2.0.4

Supported Platforms on AIX

Table 6, "IBM AIX 5L 5.3 on PowerPC Platforms" lists the application servers, JDKs, and backend databases supported by SOA Systinet on IBM AIX 5L 5.3 on PowerPC platforms.

Table 6. IBM AIX 5L 5.3 on PowerPC Platforms

J2EE Application Server	Java JDK	Relational Database
IBM WebSphere® 6.1.0.23	built-in JDK	DB2 9.1.5

Supported Platforms on HP-UX

Table 7, "HP-UX 11.31 on Itanium Platforms" lists the application servers, JDKs, and backend databases supported by SOA Systinet on HP-UX 11.31 on Itanium platforms.

Table 7. HP-UX 11.31 on Itanium Platforms

J2EE Application Server	Java JDK	Relational Database
BEA® WebLogic Server ® 9.2 MP3	HP JDK 1.5	Oracle 10.2.0.4

J2EE Application Server	Java JDK	Relational Database
JBoss® EAP 4.3.0	HP JDK 1.5	Oracle 10.2.0.4



If you use JBoss with HP-UX, there is a known JVM bug that results in a `ClassCircularityError` in HP-UX when SOA Systinet starts.

Avoid this error by setting the `shared.as.jboss.preloading.classes.at.startup` property during installation. See [Using the GUI Installer on page 19](#) or [Chapter 4, Deploying SOA Systinet](#).

For details of the issue, see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4699981 and http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4670071.

Supported Databases

For each supported database, you require the appropriate driver.

Table 8. Supported Databases and Drivers

Database	DB Version	Supported Drivers	Driver Version	Driver Class
Oracle Database	10.2.0.4 or higher	ojdbc14.jar, orai18n.jar	10.2.0.4 or higher	oracle.jdbc.driver.OracleDriver
IBM DB2	9.1.5 or higher	db2jcc.jar, db2jcc_license_cu.jar	3.2 or higher	com.ibm.db2.jcc.DB2Driver
Microsoft SQL Server	2005 (SP2) 9.00.3042 or higher	sqljdbc.jar	1.2 or higher	com.microsoft.sqlserver.jdbc.SQLServerDriver

Supported Product Integrations

SOA Systinet integrates with the following products and versions:

Table 9. Supported Product Integrations

Product	Version	Features
UDDI Registry	v2	Import/export/synchronize data.
	v3	Import/export/synchronize data. Import/Export taxonomies (HP SOA Registry Foundation only).
HP Business Availability Center	7.5	uCMDB service discovery. Service entry to governance/synchronization. BAC health report monitoring (synthetic data only).
HP SOA Policy Enforcer	3.10	Shared service performance monitoring.
HP Service Test Manager	9.41	Service test monitoring (requires Service Test Add-in for Quality Center 9.50 for QC clients).

Designing Your Deployment

HP SOA Systinet can be deployed on a wide range of scales. You must design your deployment to match the scale of your network and your own J2EE application installation procedures. Broadly speaking, there are the following types of deployment:

- [Development on page 16](#)

To evaluate the product, you can deploy it on a single machine.

- [Production on page 16](#)

To use SOA Systinet in a production environment, cluster it with SOA Systinet installed on multiple nodes.

- [Upgrade on page 17](#)

To upgrade from an earlier version of SOA Systinet, install version 3.20, apply any extensions, and migrate your data.

Development

If you are a developer, CIO, or other IT manager who wants to learn the functions of SOA Systinet, this is the correct type of deployment for you. It should be on one machine and preferably on one J2EE server instance. The simplest approach is to deploy SOA Systinet to the JBoss application server.

Use the installation wizard to deploy the product to JBoss, following the default settings. Server configuration for JBoss is handled within this wizard and in the `serverstart` and `serverstop` scripts.

If you use an application server other than JBoss, the installation wizard creates an EAR file, which you then deploy using the application server tools. You must also modify server classpaths, configure JMS and set Java properties yourself.

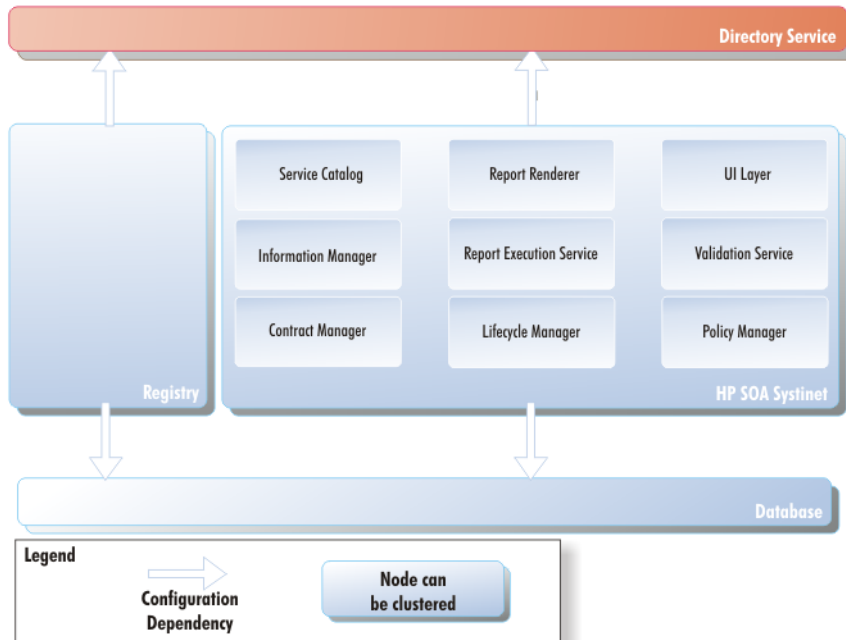
For installation details, see [Chapter 2, Basic Installation](#).

Production

Deploying SOA Systinet for use in a production environment is complex. SOA Systinet is likely to be clustered and linked to a database and directory service on separate machines. A schematic of a production deployment is shown in [Figure 1, “Production Environment Deployment”](#). If you are creating such a deployment, you should already have a set of tools and procedures for deploying J2EE applications and managing relational databases.

For details, see [Chapter 3, Setting Up Production Environments](#).

Figure 1. Production Environment Deployment



When you deploy SOA Systinet to a production environment, you may need additional configuration options that are not available in the GUI installer.

You can install SOA Systinet using the command line.

For details, see [Chapter 4, Deploying SOA Systinet](#).

Upgrade

If you have an earlier installation of SOA Systinet (2.52, 3.00, and 3.10), you can upgrade to SOA Systinet 3.20.

To upgrade from an earlier SOA Systinet version:

- 1 Install SOA Systinet 3.20 according to your deployment requirements.
- 2 If you have customized extensions, apply them to SOA Systinet 3.20:
 - a Install HP SOA Systinet Customization Editor as part of HP SOA Systinet Workbench 3.20.
For details, see the *HP SOA Systinet Customization Editor Guide*.
 - b Open the old extensions in Customization Editor 3.20.
For details, see the *HP SOA Systinet Customization Editor Guide*.
 - c Build the extensions in Customization Editor 3.20.
For details, see the *HP SOA Systinet Customization Editor Guide*.
 - d Apply the extensions to SOA Systinet 3.20.
For details, see "Applying Extensions" in the *HP SOA Systinet Administrator Guide*.
- 3 Migrate your data from earlier SOA Systinet versions to SOA Systinet 3.20.
For details, see "Data Migration" in the *HP SOA Systinet Administrator Guide*.

2 Basic Installation

Using the GUI Installer is the easiest way to install SOA Systinet. However, it may not be suitable for all the configuration options required by production environments.

Production installation and deployment is described in detail in [Chapter 4, Deploying SOA Systinet](#).

If you want to use the GUI installer for a production environment or for a non-JBoss application server, you should read [Chapter 3, Setting Up Production Environments](#) before running the installation.

This chapter contains the following sections:

- [Using the GUI Installer on page 19](#)
- [GUI Installation Options on page 26](#)
- [After GUI Installation on page 34](#)

Using the GUI Installer

Before using the GUI Installer, make sure that you have a correctly set up environment.

For hardware and software requirements, as well as supported platforms, see [Chapter 1, Getting Started](#).

For an evaluation environment, you need valid credentials to a configured database. For details, see [Preparing Databases on page 40](#).

JBoss does not require any additional configuration for evaluation purposes. If you are using the GUI installation for a production environment with JBoss or for a different application server, see [Setting Up Application Servers on page 52](#).

To install SOA Systinet using the GUI Installation wizard:

- 1 Make sure the application server is not running.

2 Do one of the following:

- Execute the file `hp-soa-systinet-3.20.jar`, located on the installation CD or in your distribution directory.
- Execute the following command:

```
java -jar hp-soa-systinet-3.20.jar
```



For JBoss with HP-UX, execute **java -jar hp-soa-systinet-3.20.jar -Dshared.as.jboss.preloading.classes.at.startup=true**.

Installation with some DB2 JDBC drivers do not manage explicit SQL commit commands in auto-commit mode. If commit errors occur during installation with DB2, execute **java -jar hp-soa-systinet-3.20.jar -Dshared.installer.db.exclude.commits=true** instead.



Additional command line options are available.

For details, see [Installation Command Line Options on page 124](#).

The GUI Installation wizard opens displaying the Welcome page.



If you have multiple JDKs installed, the first JDK in the path is used.

To ensure that the correct JDK is used:

- 1 Open a command prompt (cmd in Windows) or a terminal session (UNIX/Linux).
- 2 Execute **echo %JAVA_HOME%** (Windows) or **echo \$JAVA_HOME** (UNIX/Linux)
- 3 Do one of the following:

- If JAVA_HOME points to JDK 1.5 then proceed with installation.
- If JAVA_HOME does not point to JDK 1.5 then reset the JAVA_HOME environment variable to a valid JDK 1.5.

- 4 Verify that JAVA_HOME is set correctly.
- 5 Execute "%JAVA_HOME%/bin/java" -jar hp-soa-systinet-3.20.jar (Windows) or "\$JAVA_HOME/bin/java" -jar hp-soa-systinet-3.20.jar (UNIX/Linux).

- 3 In the Welcome page, review the hardware and software requirements, and then click **Next**.

The License page opens.

- 4 In the License page, review the license. The License page shows the license in English, German, Spanish, and French.

Click **Show the license agreement in more languages** to open a PDF which also contains the license agreement in Japanese, Korean, Chinese, and Taiwanese.

Select **I Accept the Terms of the License Agreement**, and then click **Next**.

The Installation Folder page opens.

- 5 In the Installation Folder page, **Browse** for the location you want to use as your SOA Systinet installation folder, and then click **Next**.

The installer unpacks the distribution files to the chosen location.

The Scenario Selection page opens.



In this document, the installation location is referred to as SOA_HOME.

- 6 In the Scenario Selection page, select **Default**, and then click **Next**.

The License Information page opens.



The **Advanced** scenarios enable you to perform parts of the installation separately. These functions are duplicated by the Setup Tool and are discussed as administration functions.

For details, see "Setup Tool" in the *HP SOA Systinet Administrator Guide*.

- 7 In the License Information page, do one of the following and then click **Next**:

- Select **Install a 60 day evaluation license**.
- Select **Enter license details** and type the license details provided by your sales representative.

The Updates page opens.



The administrator can change the license at a later date.

For details, see "Managing the License" in the *HP SOA Systinet Administrator Guide*.

- 8 In the Updates page, use **Add** and **Remove** to select updates to apply during installation, and then click **Next**.

The Custom Extensions page opens.

- 9 In the Custom Extensions page, use **Add** and **Remove** to select extensions to apply during installation, and then click **Next**.

The Password Encryption page opens.

- 10 In the Password Encryption page select whether SOA Systinet protects credentials for access to other systems with strong encryption:

- For production or sensitive installations, select **Enable**, type the Master Passphrase and Confirm Passphrase, and then click **Next**.
- For demo installations, select **Disable**, and then click **Next**.

The installer validates the encryption and the Database Selection page opens.



After installation with encryption, all passwords stored in the configuration file are in an encrypted form unreadable without the provided passphrase. For executions of the setup tool and some other command line tools you may need to enter passphrase or provide it using the **--passphrase** command line option.

- 11 In the Database Selection Page, select the database type to use, and then click **Next**.

The Database Setup Operations page opens.

- 12 In the Database Setup Operations page, do one of the following:

- Select **Create Database**, and click **Next**.
- Select **Create Schema**, and click **Next**.

A different Database Options page opens, depending on the database and whether you are creating a database or a schema.

For more details about database set up options, see [Database Installation Types on page 41](#).

- 13 In the Database Options page, type the parameters you want, and then click **Next**.

For parameter details, see the following sections:

- [DB2 Create Database on page 27](#)
- [DB2 Create Schema on page 28](#)
- [MSSQL Database Parameters on page 29](#)
- [Oracle Create Database on page 30](#)

- [Oracle Create Schema on page 31](#)

The JDBC Drivers page opens.

- 14 In the JDBC Drivers page, **Browse** to locate the driver to use, and then click **Next**.



Separate multiple driver names with commas.

If you are using Oracle Database, use `ojdbc14.jar`. If you use localization you may also require `orai18n.jar`.

The installer validates the database parameters, the configuration tables, and the driver.

The Repository Import page opens.

- 15 In the Repository Import page, do one of the following:

- Select **Initial Import**, and then click **Next**.
- Select **Custom Import**, **Browse** for a custom image, and then click **Next**.

The installer validates the data image.

The Application Server Selection page opens.

- 16 In the Application Server Selection page, select the application server to use and click **Next**.

For JBoss, the JBoss Deployment Properties page opens.

For other applications servers, skip to [Step 18](#), the Endpoint Properties page.



For evaluation purposes, HP Software recommend JBoss. Other application servers require additional set up and configuration.

For details of production deployment, see [Chapter 4, Deploying SOA Systinet](#).

- 17 In the JBoss Deployment Properties page, **Browse** for the JBoss application server installation folder and deployment directory, and then click **Next**.

The installer verifies the data source and the JBoss settings.

The Endpoint Properties page opens.

- 18 In the Endpoint Properties page, specify the endpoint properties, and then click **Next**.

The User Management Integration page opens.



If you change the port numbers from their default values, you must also change the application server configuration to use these ports.

For JBoss details, see [Configuring JBoss Port Numbers on page 64](#).

- 19 In the User Management Integration page do one of the following:

- To store accounts in your database, do not select **LDAP**, click **Next**, and continue to [Step 21](#).
- To integrate with an LDAP server account store, select **LDAP** and click **Next**.

The installer continues with LDAP settings as described in [LDAP Options on page 32](#) and then returns to [Step 20](#).

- 20 Enter the system mail account to be used as the source of automatic notification mails and system messages, and then click **Next**.

The Administrator Account Configuration page opens.

- 21 In the Administrator Account Configuration page, set the administrator credentials, and then click **Next**.

The SMTP Server Authentication page opens.



The administrator login name must be valid for the selected application server instance. The user with the specified name becomes an SOA Systinet administrator. For JBoss the specified administrator account is automatically created.

- 22 If you want mail notifications set the server and email.

To authenticate, select **Authenticate** and enter the SMTP server credentials.

Click **Next** to continue.

The installer creates the client package and the Confirmation page opens.

- 23 In the Confirmation page, click **Next** to start the installation process.

The Installation Progress page opens.

- 24 In the Installation Progress page, track each step of the installation. When the installation is complete, click **Next**.

The Installation Finished page opens.

- 25 Click **Finish** to exit the Installation Wizard.



You can review the installation and configuration settings in the log file:

```
SOA_HOME/logs/install.log
```

Depending on the installation options, you may need to perform some additional operations after installing SOA Systinet.

For details, see [After GUI Installation on page 34](#).

GUI Installation Options

During GUI Installation, the pages shown may vary depending on the options you choose.

This section contains the following sections:

- [Database Installation Parameters on page 27](#)
- [LDAP Options on page 32](#)

Database Installation Parameters

During GUI Installation, you select a database operation to perform and a database type.

The Database Options page varies depending on the choices you make.

For details, see the following sections:

- [DB2 Create Database on page 27](#)
- [DB2 Create Schema on page 28](#)
- [MSSQL Database Parameters on page 29](#)
- [Oracle Create Database on page 30](#)
- [Oracle Create Schema on page 31](#)

DB2 Create Database

To create a new tablespace in DB2, set the parameters listed in [Table 10, "DB2 Create Tablespace Parameters"](#).

Table 10. DB2 Create Tablespace Parameters

Parameter	Description	Notes
Database Server Address	Hostname or IP address where the database server is accessible.	In the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the hostname is <code>dbhost42</code> .
Database Server Port	Connection port for the database.	In the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the port number is <code>50000</code> .

Parameter	Description	Notes
Existing Database Name	Name of the database.	In the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the database name is <code>platform</code> .
Database Administrator Name	User name and password of the administrator of the database.	—
Database Administrator Password		
New Database Tablespace	Name of the tablespace to create	The tablespace name must not conflict with existing objects in the database.
Tablespace Datafile	Path to the tablespace datafile that is stored on the database host machine.	
Existing Database User Name	Name and password of an existing database user.	—
Database User Password		
Buffer Pool /with 32k page size/	Buffer pool to use for the tablespace.	

For more details, see [Setting Up IBM DB2 on page 46](#).

DB2 Create Schema

To create a new schema in DB2, set the parameters listed in [Table 11, "DB2 Create Schema Parameters"](#).

Table 11. DB2 Create Schema Parameters

Parameter	Description	Notes
Database Server Address	Hostname or IP address where the database server is accessible.	For example, in the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the hostname is <code>dbhost42</code> .

Parameter	Description	Notes
Database Server Port	Connection port for the database.	For example, in the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the port number is 50000.
Existing Database Name	Name of the database.	For example, in the database connection string <code>jdbc:db2://dbhost42:50000/platform</code> , the database name is <code>platform</code> .
Existing Database User Name	User name and password of an existing database user.	—
Database User Password		
Database Tablespace		

For more details, see [Setting Up IBM DB2 on page 46](#).

MSSQL Database Parameters

To create a new tablespace or schema in MSSQL, set the parameters listed in [Table 12, "MSSQL Database Parameters"](#).

Table 12. MSSQL Database Parameters

Parameter	Description	Notes
Database Server Address	Hostname or IP address where the database server is accessible.	For example, in the database connection string <code>jdbc:sqlserver://sqlhost:1433:platform</code> , the hostname is <code>sqlhost</code> .
Database Server Port	Connection port for the database.	For example, in the database connection string <code>jdbc:sqlserver://sqlhost:1433:platform</code> , the port number is 1433.
New Database Name	Name of the database.	For example, in the database connection string

		jdbc:sqlserver://sqlhost:1433:platform, the database name is platform.
Parameter	Description	Notes
Existing Database User Name	For the Create Schema option the user must have schema creation rights. For the Create Database option the user must have the database creator role.	—
Database Administrator Password		

For more details, see [Setting Up Microsoft SQL](#) on page 49.

Oracle Create Database

To create a new tablespace in Oracle, set the parameters listed in [Table 13, "Oracle Create Database Parameters"](#).

Table 13. Oracle Create Database Parameters

Parameter	Description	Notes
Database Server Address	Hostname or IP address where the database server is accessible.	For example, in the database connection string jdbc:oracle:thin:@orahost:1521/platform, the hostname is orahost.
Database Server Port	Connection port for the database.	For example, in the database connection string jdbc:oracle:thin:@orahost:1521/platform, the port number is 1521.
Existing Database Name	Name of the database.	For example, in the database connection string jdbc:oracle:thin:@orahost:1521/platform, the database name is platform.
Full Connection String	Full connection string to the database.	Select this as option as an alternative to inputting the individual connection parameters.

Parameter	Description	Notes
Database Administrator Name	User name and password of the administrator of the database.	—
Database Administrator Password		
New Database Tablespace	Name of the tablespace to create.	The tablespace name must not conflict with existing objects in the database.
Tablespace Datafile	Path to the tablespace datafile that is stored on the database host machine.	The new database tablespace must not conflict with existing objects in the database.
New Database User Name	Name and password of a new database user.	The user name must not conflict with existing objects in the database.
Database User Password		
Confirm Password		

For more details, see [Setting Up Oracle Database on page 43](#).

Oracle Create Schema

To create a new schema in Oracle, set the parameters listed in [Table 14, "Oracle Create Schema Parameters"](#).

Table 14. Oracle Create Schema Parameters

Parameter	Description	Notes
Database Server Address	Hostname or IP address where the database server is accessible.	For example, in the database connection string jdbc:oracle:thin:@orahost:1521/platform, the hostname is orahost.
Database Server Port	Connection port for the database.	For example, in the database connection string jdbc:oracle:thin:@orahost:1521/platform, the port number is 1521.

Parameter	Description	Notes
Existing Database Name	Name of the database.	For example, in the database connection string <code>jdbc:oracle:thin:@orahost:1521/platform</code> , the database name is <code>platform</code> .
Full Connection String	Full connection string to the database.	Select this as option an alternative to inputting the individual connection parameters.
Database Administrator Name	User name and password of the administrator of the database.	—
Database Administrator Password		

For more details, see [Setting Up Oracle Database on page 43](#).

LDAP Options

If you select **LDAP** in the User Management page of GUI Installation, the installer continues with LDAP service settings.

For more details about LDAP integration, see [LDAP Accounts Integration on page 118](#).

To set LDAP options:

- 1 In the LDAP Service page, set the LDAP connection parameters, credentials, and case sensitivity, and then click **Next**.
- 2 In the LDAP Search Rules page enter the parameters described in [Table 15, "LDAP Search Rules Properties"](#), and then click **Next**:

Table 15. LDAP Search Rules Properties

Property	Description
Search Filter	The notation of the search filter conforms to the LDAP search notation. You can specify the LDAP node property that matches the user account or group.

Property	Description	
Search Base	LDAP is searched from this base according to the Search Scope settings.	
Search Scope	One-level Scope	Only direct sub-nodes of the search base (entries one level below the search base) are searched. The base entry is not included in the scope.
	Subtree Scope	The search base and all its sub-nodes are searched.
Results Limit	Number of items returned when searching LDAP. If more results are returned by an LDAP search the remainder are disregarded and not shown.	

- 3 In the User Property Mapping page, use **Add** and **Remove** to set the property mappings, and then click **Next**.

The Group Properties page opens.

You must map the following mandatory user account properties from an LDAP server:

```
java.lang.String loginName
java.lang.String fullName
```

You can map the following optional user account properties from an LDAP server:

```
java.lang.String Email
java.lang.String Description
java.lang.String LanguageCode
java.lang.String Phone
java.lang.String AlternatePhone
java.lang.String Address
java.lang.String City
java.lang.String Country
```

- 4 In the Group Properties page, enter the parameters described in [Table 15, "LDAP Search Rules Properties"](#), and then click **Next**.

The Group Property Mapping page opens.

- 5 In the Group Property Mapping page, use **Add** and **Remove** to set the property mappings, and then click **Next**.

This is the last LDAP options page and the installer continues with the Administrator Account Configuration page.

The following mandatory group properties must be mapped from an LDAP server:

```
java.lang.String name  
java.lang.String member
```

The following optional group properties can be mapped from an LDAP server:

```
java.lang.string Owner  
java.lang.String Description
```



Ensure that your mappings are correct and that these properties exist on your LDAP server. The incorrect mapping of any properties, even optional ones, can have a severe performance impact for sign-in for some LDAP services.



SOA Systinet log-ons are case-sensitive by default. If you want the log-on name to be case-insensitive you must add the following property to `SOA_HOME/conf/setup/configuration-properties.xml`:

```
shared.um.account.caseInsensitiveLoginName=true
```

You must also ensure that the application server uses matching case sensitive or insensitive authentication as well.

After GUI Installation

Depending on your deployment environment and intended use, there may be a number of steps required to complete your SOA Systinet deployment.

If you used the GUI installer with a non-JBoss application server, see [Deploying the EAR File on page 143](#) for EAR deployment instructions.

If you used the GUI installer for a production deployment, see [Chapter 5, After Installation](#) for a list of additional configuration procedures that may be required.

To start the SOA Systinet server, do one of the following:

- For JBoss execute the following command:

```
SOA_HOME/bin/serverstart
```

- For other application servers, use the relevant server start procedure.

3 Setting Up Production Environments

Before installing SOA Systinet, you must prepare your environment.

Set up your environment as described in the following sections:

- [SOA Systinet Self-Test on page 37](#)
- [Preparing Databases on page 40](#)
- [Setting Up Application Servers on page 52](#)
- [LDAP Accounts Integration on page 118](#)
- [Siteminder Integration on page 120](#)

SOA Systinet Self-Test

Self-Test is a tool that checks various aspects of deployment. It can be used during the setup of particular resources on an application server such as data sources, JNDI, and JMS which are required for the successful deployment of SOA Systinet.

The following sections describe the deployment and use of Self-Test:

- [Stand-Alone Self-Test Deployment on page 38](#)

HP Software recommend deploying Self-Test as a standalone application before you start to set up your application server for SOA Systinet deployment.

- [Running Self-Test on page 38](#)

You can run Self-Test during application setup or after installation.

- [Turning Off Self-Test Output on page 40](#)

As a security measure you can turn off the Self-Test output.

Stand-Alone Self-Test Deployment

The package is prepared as a standalone application for deployment to application servers.

To deploy Self-Test as a standalone application:

- 1 Extract the SOA Systinet installer archive with the following command:

```
java -jar hp-soa-systinet-3.20.jar -x SOA_HOME
```

The Self-Test application package is `SOA_HOME/deploy/self-test-standalone.war`.

- 2 Deploy the WAR file using the functionality of your application server or copy the WAR file to your JBoss deploy directory.



If you set password encryption after deploying the Self-Tester during installation or with the setup tool, you must redeploy the WAR.

Running Self-Test

To execute the stand-alone self-tester and access its output, start the Self-Test application in your application server and then access the following URL:

```
http://hostname:port/self-test-standalone
```



`hostname:port` should match your application server.

The self-tester performs the following checks:

Table 16. Self-Tests

Self-Test	Description
Product configuration checks	Checks product configuration, versions, and libraries.

Self-Test	Description
Product runtime checks	Checks logging configuration, and outputs product base URLs.
Application server checks	Checks application server and JVM settings.
JNDI checks	Checks required JNDI resources.
Datasource checks	Checks the data source connection.
JMS checks	Checks the sending of JMS messages to required JMS destinations.
LDAP checks	Checks LDAP connectivity, if configured during installation or setup.
Performance	Basic SOA Systinet performance checks.

You can view the self-test results in the server output console or with your browser.

In the default configuration, the server console output includes only information about the groups of checks that are run and any errors that occur. The full self-test output is stored in the application server log folder, `systinet_self_test.log`.

The web output is more informative and readable, showing all the checks run and the results.

Access the standalone self-test output at the following URL:

`http://hostname:port/self-test-standalone`

If errors occur, the self-tester provides details about the errors and suggests how to solve the underlying problems.

After installation, Self-Test is also available from the Administration menu in the Tools tab as part of the SOA Systinet EAR and opens URL: `http://hostname:port/context/self-test`.

Self-test also enables you to test HTTP/HTTPS connections to simulate access to external resources in the same way as a deployed SOA Systinet. Access this feature at the following URL:

`http://hostname:port/context/self-test/self-http-test`

During application setup and deployment, HP Software recommend running self-test at the following milestones:

- Before starting application server setup. At this point only the Application server checks should pass.

- After setting up JDBC resources. At this point the Datasource checks should pass if the application server is configured correctly.
- After setting up JMS resources. At this point the JMS checks should pass if the application server is configured correctly.
- After creating mail sessions. At this point the JNDI checks should pass if the application server is configured correctly.
- After deploying the SOA Systinet EAR file and starting SOA Systinet. At this point all checks should pass if the application server and SOA Systinet are configured correctly.

Turning Off Self-Test Output

The self-test output is accessible to anyone using the URL. For security reasons, you can switch off access to the self-test output after a completed deployment of SOA Systinet passes the self-test.

To switch off the self-test output:

- 1 Sign in to SOA Systinet as the administrator.
- 2 In the Tools tab Administration menu, click **Configuration**.
The Configure page opens.
- 3 In the Configure page, deselect **Self Test Access**.
- 4 Click **Save**.

To disable the standalone self-tester, undeploy the `self-test-standalone.war` package from your server.

To verify that the self-test has been disabled, check the self-test output URL.

Preparing Databases

This section describes database administration tasks for SOA Systinet. The database administrator must perform tasks at the time of installation and may also have tasks when SOA Systinet is updated, extensions are applied, or data is migrated.

Before you can install SOA Systinet the database administrator must set up the database.

Read [Database Installation Types on page 41](#) first for information about the different database installation scenarios which vary according to the required level of access to the database.

The database specific sections describe database specific prerequisites and procedures describing how to create the various user types required by the different database installation scenarios.

- [Setting Up Oracle Database on page 43](#)
- [Setting Up IBM DB2 on page 46](#)
- [Setting Up Microsoft SQL on page 49](#)

Database Installation Types

There are several ways to arrange the database and schema for SOA Systinet.

The option required is usually determined by the level of access to the database.

The options are described in the following sections:

- [Create Schema on page 41](#)
- [Create Database on page 42](#)
- [Manual Database Arrangement on page 42](#)

After installation, there are more database options available using the Setup Tool.

For details, see "Setup Tool" in the *HP SOA Systinet Administrator Guide*.

Create Schema

The Create Schema option, available in the GUI installer and command-line deployment, creates tables and indexes in the default schema in an existing database or tablespace provided by the database administrator. Select this method if you have an account in a database with an empty schema (recommended) and privileges to create tables and indexes.



In this document, *power user* refers to users with the privilege to create tables and indexes.

Create Database

The option to create a database is available in the GUI installer and command-line deployment. This option automates database arrangement as much as possible, but requires database administrator credentials. The process creates users, the database or tablespace depending on your database type, and continues with the creation of the schema.

There are some differences in the create database process depending on the database type:

- **IBM DB2**

This option requires an existing database, OS user and database administrator credentials.

This option does not create a new physical database. It creates a tablespace in an existing database to separate repository data. The user is then granted privileges to use the tablespace, create tables, and connect to the database.

- **Microsoft SQL**

This option requires an existing user with the database creator role.

This option creates a new physical database with collation inherited from the server settings.

- **Oracle Database**

This option requires an existing database and database administrator credentials.

This option does not create a new physical database. It creates a new tablespace to hold SOA Systinet data separately and creates a new database account which uses the new tablespace as its default tablespace.

Manual Database Arrangement

The database administrator may want to arrange the database manually:

- In some cases, the database administrator (DBA) cannot share the DBA credentials required for the Create Database option or the *power user* credentials for the Create Schema option.
- In some cases, the database administrator may want to amend the default DDL scripts. For example, to create indexes in a separate tablespace.

In these cases, the database administrator must perform the database related installation operations manually as part of *Decoupled Database Installation*.

For details, see [Decoupled Database Deployment on page 140](#).

Typically the database administrator creates a power user account for the SOA Systinet schema and a common user account with minimal privileges to insert, select, update, and delete SQL operations in power user tables.

The database administrator does not distribute the power user credentials and provides the common user credentials to the SOA Systinet administrator to configure the application server datasource.

Setting Up Oracle Database

Configure the Oracle database as follows for use with SOA Systinet:

- If you are clustering Oracle database (RAC), you must use Oracle Database 10.2.0.4 or higher. SOA Systinet does not support RAC in earlier versions.
- SOA Systinet installation requires a JDBC driver, use `ojdbc14.jar`. If you use localization you may also require `orai18n.jar`.
- To use SOA Systinet Full Text Search, include the "Oracle Text" extension when installing the Oracle server. The "Oracle Text" extension is applied to Oracle by default.
- HP Software strongly recommend creating a database that uses the Unicode for Database Character Set (NLS_CHARACTERSET=AL32UTF8). If you use a non-Unicode database, you may encounter problems storing and searching some national characters outside your character set. Changing the character set after installation is only possible by creating a new database.
- HP Software recommend setting the `cursor_sharing` parameter to `FORCE` to improve performance and economize shared pool usage.

- Create accounts based on the database installation type selected for SOA Systinet installation.

For details, see [Database Installation Types on page 41](#).

The access required is defined by the database installation type:

- For the Create Database option an account is created by the installer.
- For the Create Schema option, if you want to separate the SOA Systinet data (recommended), create a tablespace in the database. Create a power user to own the schema, with the new tablespace as its default tablespace.

For details, see [Setting Up an Oracle Power User on page 44](#).

- For Manual Database Arrangement, create a tablespace in the database, create a power user account to own the schema, with the new tablespace as its default tablespace. Create a *common user* account with minimal privileges.

For details, see [Setting Up an Oracle Power User on page 44](#) and [Setting Up an Oracle Common User on page 45](#).



After installation, if you require SOA Systinet Full Text Search to be enabled, set up Oracle indexing.

For details, see [Enabling Full Text Search on Oracle on page 173](#).

Setting Up an Oracle Power User

In order to use the Create Schema option during installation or for Manual Database Arrangement, the database administrator should create a *power user* with appropriate privileges to the database.

To set up a power user on Oracle:

- 1 HP Software recommend creating a new tablespace to hold SOA Systinet data.
- 2 Create an account that can create schema items, with the new tablespace as its default tablespace.
- 3 Grant privileges to the account to connect to the database and create tables, indexes, and sequences.

4 Optionally, grant the account the privilege to execute "CTXSYS"."CTX_DDL."

This privilege is a precondition for using the SOA Systinet full-text search feature on the database.

Setting Up an Oracle Common User

In cases where the database administrator restricts access to the database to just select, insert, update, and delete operations, SOA Systinet requires a user with these privileges.



The SOA Systinet schema must exist before you create the common user.

To set up a common user on Oracle:

1 Save the following SQL statements to the `script.sql` file:

```
set pagesize 0;
set pagesize 0;
set line 200;
set verify off
set feedback off
spool ./grant.sql
SELECT 'GRANT INSERT, UPDATE, DELETE, SELECT ON ' || table_name || ' TO &2;' FROM user_tables;
SELECT 'GRANT SELECT ON ' || sequence_name || ' TO &2;' FROM user_sequences;
spool off
spool ./synonyms.sql
SELECT 'CREATE SYNONYM ' || table_name || ' FOR &1' || '.' || table_name || ';' FROM user_tables;
SELECT 'CREATE SYNONYM ' || sequence_name || ' FOR &1' || '.' || sequence_name || ';' FROM
user_sequences;
spool off
```

These statements generate scripts to set the environment, grant rights and create synonyms.

2 Connect to the database as the `power_user` and execute `script.sql` to produce the scripts `grant.sql` and `synonyms.sql`. Then execute `grant.sql`.

```
sqlplus power_user/password@SID
-- generate grant and create synonym statements
@script.sql power_user common_user
-- execute grant.sql
@grant.sql
```

```
exit
```

- 3 As the `common_user`, execute `synonyms.sql`.

```
sqlplus common_user/password@SID
-- execute synonym.sql
@synonyms.sql
exit
```

Oracle with WebSphere

Configure the Oracle Database to support WebSphere with XA transactions over Oracle datasources.

As user SYS, run the following commands on your Oracle server:

```
grant select on pending_trans$ to public;
grant select on dba_2pc_pending to public;
grant select on dba_pending_transactions to public;
grant execute on dbms_system to <user>;
```

Setting Up IBM DB2

You can use SOA Systinet with an IBM DB2 database. The database requires set up and configuration prior to installing SOA Systinet.

To configure the DB2 database for use with SOA Systinet:

- 1 If you plan to use the SOA Systinet full text search feature, make sure the optional DB2 Net Search Extender is installed.
- 2 SOA Systinet installation requires a JDBC driver, use `db2jcc.jar` and `db2jcc_license_cu.jar`.
- 3 If one does not exist, create a database that uses the UTF-8 Code Set.
- 4 If it does not exist, create a bufferpool with a page size of 32k. Create a *system temporary tablespace* with 32k page size, using the bufferpool.
- 5 To ensure the successful import or export of large data images HP Software recommend increasing the log file size (`LOGFILSIZ`) parameter to 2048 or higher and the number of primary log files (`LOGPRIMARY`) parameter to 15 or higher.

To ensure that there is sufficient memory HP Software recommend increasing the application heap size (*APPLHEAPSZ*) parameter to 512 or higher.

- 6 Increase value of the *stmtheap* property to 16400 or more to avoid following exception:

```
com.ibm.db2.jcc.b.SqlException: DB2 SQL error: SQLCODE: -101, SQLSTATE: 54001, SQLERRMC:  
null
```

- 7 Create an OS user account to hold the SOA Systinet data.
- 8 Create accounts based on the database installation type.

For details, see [Database Installation Types on page 41](#).

- For the Create Database option no additional accounts are required.
- For the Create Schema option, create a power user.

For details, see [Setting Up a DB2 Power User on page 47](#) .

- For Manual Database Arrangement, create a power user account to own the schema, and create a common user account with minimal privileges.

For details, see [Setting Up a DB2 Power User on page 47](#) and [Setting Up a DB2 Common User on page 48](#).



After installation, if you require SOA Systinet Full Text Search to be enabled, set up indexes on DB2.

For details, see [Enabling Full Text Search on DB2 on page 169](#).

Setting Up a DB2 Power User

To use the Create Schema option during installation or Manual Database Arrangement, the database administrator should create a *power user* with appropriate privileges to the database.

To set up a power user on DB2:

- 1 Create a tablespace using the 32k page-sized bufferpool to hold SOA Systinet data.
- 2 Grant CONNECT, CREATETAB, and IMPLICIT_SCHEMA privileges to the user account.
- 3 Grant use of the tablespace to the OS user account.

Setting Up a DB2 Common User

In cases where the database administrator restricts access to the database to just select, insert, update, and delete operations, SOA Systinet requires a user with these privileges.



The SOA Systinet schema must exist prior to creating the common user.

To set up a common user on DB2:

- 1 Create an OS account for the common user.
- 2 Grant the common user connection privileges:

```
GRANT CONNECT ON DATABASE TO common_user
```
- 3 Open the DB2 Command Editor and connect to the database using *power user* credentials.
- 4 Generate a list of commands granting privileges to database tables to the common user with the following command:

```
SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE ' || TABNAME || '  
      TO common_user;' FROM syscat.tables  
      WHERE LOWER(tabschema) = LOWER('power_user')
```

- 5 After the results display, click **Fetch More Rows** at least twice until all rows are displayed.
- 6 Select all the resulting commands and copy them to the clipboard.
- 7 In the Commands window, paste the clipboard contents.

- 8 Execute the commands.
- 9 Generate a list of commands to create aliases for the tables with the following command:

```
SELECT 'CREATE ALIAS ' || TABNAME || ' FOR power_user.'  
      || TABNAME || ';' FROM syscat.tables  
WHERE LOWER(tabschema) = LOWER('power_user')
```

- 10 After the results display, click **Fetch More Rows** at least twice until all rows are displayed.
- 11 Select all resulting commands and copy them to the clipboard.
- 12 Open a new instance of DB2 Command Editor and connect to the database using the common user credentials.
- 13 In the common user Commands window, paste the clipboard contents.
- 14 Execute the commands.

Setting Up Microsoft SQL

You can use SOA Systinet with an Microsoft SQL database. The database requires set up and configuration prior to installing SOA Systinet.

- 1 Use SQL Server Configuration Manager to enable the TCP/IP protocol and use a static port (for example 1433).
- 2 SOA Systinet installation requires a JDBC driver, use `sqljdbc.jar`.
- 3 SOA Systinet requires XA transactions support.

For details about setting up XA transaction support, go to the following location:

<http://msdn2.microsoft.com/en-us/library/aa342335.aspx>

- 4 If you want to use the full-text search feature in SOA Systinet, make sure that the Full-Text Search engine is installed together with the database engine during the installation of MSSQL Server.
- 5 Create a login in the database server to hold SOA Systinet tables in the database. The login must have the *database creator* role.

The login must be able to access the master database for XA related stored procedures:

- Create a user in the master database for the login.
- Add the account to the SqlJDBCXAUser role.

6 Create users based on the database installation type selected for SOA Systinet installation.

For details, see [Database Installation Types on page 41](#).

- For the Create Database option the installer uses the login to automatically arrange the database.

The created database inherits collation from the MSSQL server default collation. HP Software recommend using case-sensitive collation, otherwise all searches are case-insensitive. Use a server with case-sensitive collation or manage database collation manually using the Create Schema option.

- For the Create Schema option, if you want to separate the SOA Systinet data (recommended), use the login to create a database with given case sensitive collation.



You can create the database on behalf of another account or use an existing account with an existing database, but you must then grant create table privileges to the new account or the existing account.

The installer uses the login to create the schema in this new database.

- For Manual Database Arrangement, use the power user login to create the database with given case sensitive collation.

Then create a common user account with minimal privileges. For details, see [Setting Up an MSSQL Common User on page 51](#).



After installation, if you require SOA Systinet Full Text Search to be enabled, set up indexes on MSSQL.

For details, see [Enabling Full Text Search on MSSQL on page 171](#).



If you intend to use user accounts in SOA Systinet that will contain non-Latin characters, you must set the default collation on the database to cover all such non-Latin characters.



To prevent some possible deadlocks, HP Software recommend executing the following statement:

```
ALTER DATABASE [database_name] SET READ_COMMITTED_SNAPSHOT ON;
```

Setting Up an MSSQL Common User

In cases where the database administrator restricts access to the database to just select, insert, update, and delete operations, SOA Systinet requires a user with these privileges.

To set up a common user on MSSQL:

- 1 Open Microsoft SQL Server Management Studio or the sqlcmd command-line editor.
- 2 Create a common user login in the server and user in the database created for SOA Systinet (systinetdb).

For example, execute the following statements:

```
USE [master]
GO
CREATE LOGIN [common_user] WITH PASSWORD=N'...', DEFAULT_DATABASE=[master],
CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
USE [systinetdb]
GO
CREATE USER [common_user] FOR LOGIN [common_user]
GO
```

- 3 Grant rights to the common user to read and write to SOA Systinet tables.

For example, execute the following statements:

```
USE [systinetdb]
GO
EXEC sp_addrolemember N'db_datawriter',N'common_user'
```

```
GO
USE [systinetdb]
GO
EXEC sp_addrolemember N'db_datareader', N'common_user'
GO
```

- 4 The login must be able to access the master database for XA related stored procedures.

Create a user in the master database for the login and add the user to the SqlJDBCXAUser role.

For example, execute the following statements:

```
USE [master]
GO
CREATE USER [common_user] FOR LOGIN [common_user]
GO
USE [master]
GO
EXEC sp_addrolemember N'SqlJDBCXAUser', N'common_user'
GO
```

Setting Up Application Servers

SOA Systinet is deployed to J2EE application servers. Each different application server must be set up prior to SOA Systinet installation.

The set up of each application server is explained in the following sections:

- [Setting Up JBoss on page 52](#)
- [Setting Up Oracle Application Server on page 72](#)
- [Setting Up WebLogic on page 83](#)
- [Setting Up WebSphere on page 99](#)

Setting Up JBoss

Deployment to JBoss requires less set up than installation to other J2EE servers. SOA Systinet installation automates deployment to JBoss. Datasources and JMS are set up on the host JBoss servers and the SOA

Systinet EAR file is deployed. The installer also creates a script for setting up the server environment and launching JBoss in simple deployment scenarios.



If you use JBoss with Windows, install it with a path that contains less than 20 characters. This limitation is caused by JBoss expanding the application in the local disk and the Windows 255 character limit on path names.

If you use JBoss with HP-UX, there is a known JVM bug that results in a `ClassCircularityError` in HP-UX when SOA Systinet starts.

Avoid this error by setting the `shared.as.jboss.preloading.classes.at.startup` property during installation. See [Using the GUI Installer on page 19](#) or [Chapter 4, Deploying SOA Systinet](#).

For details of the issue, see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4699981 and http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4670071.

You may need to modify the JBoss application server for it to host SOA Systinet in production environments.

These modifications are covered in the following sections where `JBOSS_HOME` refers to the application server installation directory, for example `JBOSS/jboss430`:

- [Configuring JMS for JBoss on page 54](#)

Configure JBoss to use the Oracle or DB2 database instead of the default HSQLDB.

- [Modifying the JBoss Run Script on page 62](#)

Configure the JBoss `run` script.

If you are using the SOA Systinet `serverstart` script, you still have to edit the memory allocation in the `run` script.

- [Setting the Datasource MaxPoolSize on page 64](#)

Increase the datasource maximum pool size for production deployments.

- [Configuring JBoss Port Numbers on page 64](#)

If you set the SOA Systinet endpoint to use ports other than the default 8080 and 8443, you must enable these ports on JBoss.

- [Deploying SOA Systinet to a JBoss Cluster on page 65](#)

Configure and deploy SOA Systinet to a JBoss cluster.

Configuring JMS for JBoss

JBoss uses JMS preconfigured for HSQLDB, which is sufficient for lightweight use in evaluation deployments. However, it has difficulty with large numbers of requests. For production deployments the JMS service should be configured to use a supported database.



SOA Systinet uses XA transactions. The application server transaction manager should be configured to have a minimum of 5 minutes for XA transaction timeout.

For details, refer to your application server documentation.

To change the JMS configuration, follow the relevant procedure for your deployment:

- [Using DB2 DS in Non-Clustered Deployments on page 54](#)
- [Using DB2 DS in Clustered Deployments on page 56](#)
- [Using MSSQL DS in Non-Clustered Deployments on page 57](#)
- [Using MSSQL DS in Clustered Deployments on page 59](#)
- [Using Oracle DS in Non-Clustered Deployments on page 60](#)
- [Using Oracle DS in Clustered Deployments on page 61](#)

Using DB2 DS in Non-Clustered Deployments

To set up JBoss JMS to use the DB2 DS in non-clustered deployments:

- 1 Copy the DB2 JDBC drivers `db2jcc.jar` and `db2jcc_license_cu.jar` to `JBOSS_HOME/server/default/lib`.
- 2 Delete the file `JBOSS_HOME/server/default/deploy/hsqldb-ds.xml` .

- 3 Copy `JBOSS_HOME/docs/examples/jca/db2-ds.xml` to `JBOSS_HOME/server/default/deploy`.
- 4 In the new copy of `db2-ds.xml`, edit the `connection-url`, `user-name`, and `password` elements to match your local environment.
- 5 Change the value of the `driver-class` element to `com.ibm.db2.jcc.DB2Driver`.
- 6 Change the value of the `jndi-name` element from `DB2DS` to `DefaultDS`.
- 7 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

Example 1. Excerpt from `db2-ds.xml`

```
<datasources>
  <local-tx-datasource>
    <jndi-name>DefaultDS</jndi-name>
    <connection-url>jdbc:db2://dbserver:50000/database</connection-url>
    <driver-class>com.ibm.db2.jcc.DB2Driver</driver-class>
    <user-name>soa_account</user-name>
    <password>soa_password</password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>15</max-pool-size>
    <metadata>
      <type-mapping>DB2</type-mapping>
    </metadata>
  </local-tx-datasource>
</datasources>
```

- 8 Save `db2-ds.xml`.
- 9 For JBoss 4.2.2, delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml` .

For JBoss EAP 4.3.0, delete the file `JBOSS_HOME/server/default/deploy/jboss-messaging.sar/hsqldb-persistence-service.xml`.

- 10 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/db2-jdbc2-service.xml` into the folder `JBOSS_HOME/server/default/deploy/jms`.

For JBoss EAP 4.3.0, copy `JBOSS_HOME/docs/examples/jms/db2-persistence-service.xml` to `JBOSS_HOME/server/default/deploy/jboss-messaging.sar`.

- 11 In the new copy of `db2-jdbc2-service.xml`, replace the string `DB2DS` with `DefaultDS`.
- 12 Save `db2-jdbc2-service.xml`.
- 13 Open `JBOSS_HOME/server/default/deploy/jms/jms-ds.xml` and set the `max-pool-size` element to the maximum number of parallel served execution requests.
- 14 Open `JBOSS_HOME/server/default/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using DB2 DS in Clustered Deployments

To set up JBoss JMS to use the DB2 DS in clustered deployments:

- 1 Copy the DB2 JDBC drivers `db2jcc.jar` and `db2jcc_license_cu.jar` to `JBOSS_HOME/server/node1/lib`.



`node1` in the path refers to a copy of the `all` configuration folder.

- 2 Delete the file `JBOSS_HOME/server/default/node1/hsqldb-ds.xml`.
- 3 Copy `JBOSS_HOME/docs/examples/jca/db2-ds.xml` to `JBOSS_HOME/server/node1/deploy`.
- 4 In the new copy of `db2-ds.xml`, edit the `connection-url`, `user-name`, and `password` elements to match your local environment.
- 5 Change the value of the `driver-class` element to `com.ibm.db2.jcc.DB2Driver`.
- 6 Change the value of the `jndi-name` element from `DB2DS` to `DefaultDS`.

- 7 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

For an example of `db2-ds.xml`, see [Step 7 in Using DB2 DS in Non-Clustered Deployments on page 54](#).

- 8 Save `db2-ds.xml`.
- 9 Delete the file `JBOSS_HOME/server/node1/deploy-hasingleton/jms/hsqldb-jdbc2-service.xml`.
- 10 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/db2-jdbc2-service.xml` into the folder `JBOSS_HOME/server/node1/deploy-hasingleton/jms`.

For JBoss EAP 4.3.0, copy `JBOSS_HOME/docs/examples/jms/db2-persistence-service.xml` to `JBOSS_HOME/server/node1/deploy-hasingleton/jboss-messaging.sar`.
- 11 In the new copy of `db2-jdbc2-service.xml`, replace the string `DB2DS` with `DefaultDS`.
- 12 Save `db2-jdbc2-service.xml`.
- 13 Open `JBOSS_HOME/server/node1/deploy/jms/hajndi-jms-ds.xml` and set the `max-pool-size` element to the maximum number of parallel served execution requests.
- 14 Open `JBOSS_HOME/server/node1/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using MSSQL DS in Non-Clustered Deployments

To set up JBoss JMS to use the MSSQL DS in non-clustered deployments:

- 1 Copy the MSSQL JDBC driver `sqljdbc.jar` to `JBOSS_HOME/server/default/lib`.
- 2 Delete the file `JBOSS_HOME/server/default/deploy/hsqldb-ds.xml` .
- 3 Copy `JBOSS_HOME/docs/examples/jca/mssql-ds.xml` to `JBOSS_HOME/server/default/deploy`.

- 4 In the new copy of `mssql-ds.xml`, edit the `connection-url`, `user-name` and `password` elements to match your local environment.
- 5 Change the value of the `jndi-name` element from `MSSQLDS` to `DefaultDS`.
- 6 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

- 7 Save `mssql-ds.xml`.
- 8 For JBoss 4.2.2, delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml` .
For JBoss EAP 4.3.0, delete the file `JBOSS_HOME/server/default/deploy/jboss-messaging.sar/hsqldb-persistence-service.xml`.
- 9 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/mssql-jdbc2-service.xml` to `JBOSS_HOME/server/default/deploy/jms`.
For JBoss 4.3.0, copy `JBOSS_HOME/docs/examples/jms/mssql-persistence-service.xml` to `JBOSS_HOME/server/default/deploy/jboss-messaging.sar`.
- 10 For JBoss 4.2.2, in the new copy of `mssql-jdbc2-service.xml`, replace the string `MSSQLDS` with `DefaultDS` and save `mssql-jdbc2-service.xml`.

- 11 Open the following file:

- For JBoss 4.2.2, `JBOSS_HOME/server/default/deploy/jms/jms-ds.xml`
- For JBoss 4.3.0, `JBOSS_HOME/server/default/deploy/jms-ds.xml`

Set the `max-pool-size` element to the maximum number of parallel served execution requests.

- 12 Open `JBOSS_HOME/server/default/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using MSSQL DS in Clustered Deployments

To set up JBoss JMS to use the MSSQL DS in clustered deployments:

- 1 Copy the MSSQL JDBC driver `sqljdbc.jar` to `JBOSS_HOME/server/node1/lib`.



`node1` in the path refers to a copy of the `all` configuration folder.

- 2 Delete the file `JBOSS_HOME/server/node1/deploy/hsqldb-ds.xml` .
- 3 Copy `JBOSS_HOME/docs/examples/jca/mssql-ds.xml` to `JBOSS_HOME/server/node1/deploy`.
- 4 In the new copy of `mssql-ds.xml`, edit the `connection-url`, `user-name` and `password` elements to match your local environment.
- 5 For Change the value of the `jndi-name` element from `MSSQLDS` to `DefaultDS`.
- 6 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

- 7 Save `mssql-ds.xml`.
- 8 For JBoss 4.2.2, delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml` .

For JBoss EAP 4.3.0, delete the file `JBOSS_HOME/server/default/deploy/jboss-messaging.sar/hsqldb-persistence-service.xml`.

- 9 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/mssql-jdbc2-service.xml` to `JBOSS_HOME/server/default/deploy/jms`.

For JBoss 4.3.0, copy `JBOSS_HOME/docs/examples/jms/mssql-persistence-service.xml` to `JBOSS_HOME/server/node1/deploy-hasingleton/jboss-messaging.sar`.

- 10 For JBoss 4.2.2, in the new copy of `mssql-jdbc2-service.xml`, replace the string `MSSQLDS` with `DefaultDS`.
Save `mssql-jdbc2-service.xml`.
- 11 Open `JBOSS_HOME/server/dnode1/deploy/jms/hajndi-jms-ds.xml` and set the `max-pool-size` element to the maximum number of parallel served execution requests.
- 12 Open `JBOSS_HOME/server/node1/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using Oracle DS in Non-Clustered Deployments

To set up JBoss JMS to use the Oracle DS in non-clustered deployments:

- 1 Copy the Oracle JDBC driver `ojdbc14.jar` to `JBOSS_HOME/server/default/lib`.
- 2 Delete the file `JBOSS_HOME/server/default/deploy/hsqldb-ds.xml`
- 3 Copy `JBOSS_HOME/docs/examples/jca/oracle-ds.xml` to `JBOSS_HOME/server/default/deploy`.
- 4 In the new copy of `oracle-ds.xml`, edit the `connection-url`, `user-name`, and `password` elements to match your local environment.
- 5 Change the value of the `jndi-name` element from `OracleDS` to `DefaultDS`.
- 6 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

- 7 Save `oracle-ds.xml`.
- 8 For JBoss 4.2.2, delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml` .

For JBoss EAP 4.3.0, delete the file `JBOSS_HOME/server/default/deploy/jboss-messaging.sar/hsqldb-persistence-service.xml`.

- 9 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/oracle-jdbc2-service.xml` to `JBOSS_HOME/server/default/deploy`.

For JBoss 4.3.0, copy `JBOSS_HOME/docs/examples/jms/oracle-persistence-service.xml` to `JBOSS_HOME/server/default/deploy/jboss-messaging.sar` .

- 10 For JBoss 4.2.2, in the new copy of the Oracle service file, replace the string `OracleDS` with `DefaultDS`.
- 11 Save the Oracle service file.
- 12 Open the following file:
 - For JBoss 4.2.2, `JBOSS_HOME/server/default/deploy/jms/jms-ds.xml`
 - For JBoss 4.3.0, `JBOSS_HOME/server/default/deploy/jms-ds.xml`

Set the `max-pool-size` element to the maximum number of parallel served execution requests.

- 13 Open `JBOSS_HOME/server/default/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.
- 14 For JBoss EAP 4.3.0, ensure that a copy of `JBOSS_HOME/server/all/lib/jgroups.jar` exists in `JBOSS_HOME/server/default/lib/jgroups.jar`. If it does not exist, then copy it.

Using Oracle DS in Clustered Deployments

To set up JBoss JMS to use the Oracle DS in clustered deployments:

- 1 Copy the Oracle JDBC driver `ojdbc14.jar` to `JBOSS_HOME/server/node1/lib`.



`node1` in the path refers to a copy of the `all` configuration folder.

- 2 Delete the file `JBOSS_HOME/server/node1/deploy/hsqldb-ds.xml` .
- 3 Copy `JBOSS_HOME/docs/examples/jca/oracle-ds.xml` to `JBOSS_HOME/server/node1/deploy`.
- 4 In the new copy of `oracle-ds.xml`, edit the `connection-url`, `user-name`, and `password` elements to match your local environment.
- 5 Change the value of the `jndi-name` element from `OracleDS` to `DefaultDS`.

- 6 Add a `max-pool-size` element at the same level as `password`, `user-name`, and `driver-class`. Set the value of `max-pool-size` to the maximum number of concurrent working users plus the number of concurrent task executions.

If you do not have an estimate of these numbers, set the `max-pool-size` to 100.

- 7 Save `oracle-ds.xml`.
- 8 For JBoss 4.2.2, delete the file `JBOSS_HOME/server/node1/deploy-hasingleton/jms/hsqldb-jdbc2-service.xml`.

For JBoss EAP 4.3.0, delete the file `JBOSS_HOME/server/node1/deploy-hasingleton/jboss-messaging.sar/hsqldb-persistence-service.xml`

- 9 For JBoss 4.2.2, copy `JBOSS_HOME/docs/examples/jms/oracle-jdbc2-service.xml` to `JBOSS_HOME/server/node1/deploy-hasingleton/jms`.

For JBoss EAP 4.3.0, copy `JBOSS_HOME/docs/examples/jms/oracle-persistence-service.xml` to `JBOSS_HOME/server/node1/deploy-hasingleton/jboss-messaging.sar`.

- 10 In the new copy of the Oracle service file, replace the string `OracleDS` with `DefaultDS`.
- 11 Save the Oracle service file.
- 12 Open `JBOSS_HOME/server/node1/deploy/jms/hajndi-jms-ds.xml` and set the `max-pool-size` element to the maximum number of parallel served execution requests.
- 13 Open `JBOSS_HOME/server/node1/deploy/jboss-web.deployer/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.
- 14 For JBoss EAP 4.3.0, ensure that a copy of `JBOSS_HOME/server/all/lib/jgroups.jar` exists in `JBOSS_HOME/server/CONFIG_NAME/lib/jgroups.jar`. If it does not exist, then copy it.

Modifying the JBoss Run Script

When you launch SOA Systinet with the `SOA_HOME/bin/serverstart` script, it sets up the JBoss environment before calling the JBoss run script. No further set up is necessary for most evaluation or development scenarios. However, `serverstart` is not appropriate for all production environments.

The following sections describes how to alter the JBoss `run` script for use in production deployments:

- [Setting awt.headless on page 63](#)
- [JBoss Memory Allocation on page 63](#)

Setting awt.headless

If JBoss is installed on UNIX, set the `java.awt.headless` property to "true."

To set java.awt.headless:

- 1 Open the `JBOSS_HOME/bin/run.bat|run.conf` script in an editor.
- 2 Insert this line where `JAVA_OPTS` is set:
`-Djava.awt.headless=true`
- 3 Save and exit the script.

JBoss Memory Allocation

Increase the maximum memory limit on the JBoss server to optimize SOA Systinet performance.

To change the memory settings:

- 1 Open the `run` script in the `bin` directory of the JBoss server. On Windows this script is `run.bat` and on UNIX systems this is `run.conf`.
- 2 Find the following lines:

```
rem JVM memory allocation pool parameters. Modify as appropriate.  
set JAVA_OPTS=%JAVA_OPTS% -Xms128m...
```
- 3 Edit the lines as follows:

```
rem JVM memory allocation pool parameters. Modify as appropriate.  
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx1300m -XX:MaxPermSize=256m
```



If you are using the `serverstart` script to launch JBoss, you can delete this line from the `JBoss run` script instead of editing it. If you leave it, however, it will overwrite the memory allocation parameters set by `serverstart`.

- 4 Save and exit the script.



Memory sizing should take performance requirements into consideration for the deployed system. These settings are only a recommendation.

Setting the Datasource MaxPoolSize

The default JBoss datasource Maximum Pool Size is not adequate for a production environment. The default MaxPoolSize based on default Oracle configuration is only 15, for example. The Maximum Pool Size should be at least 1/4 the number of parallel requests that you require to be handled simultaneously.

To increase the maximum pool size:

- 1 Open `JBOSS_HOME/server/CONFIG_HOME/deploy/hpsoasystinet-xa-ds.xml` in an editor. (`CONFIG_HOME` refers to the JBoss configuration to which you have deployed SOA Systinet. For non-clustered deployments, this is usually `default` and for clustered deployments, this is usually `all`.)
- 2 Edit the element `max-pool-size`. Its value should be at least 1/4 of the number of simultaneous parallel requests.
- 3 Save your changes and exit.

Configuring JBoss Port Numbers

By default, SOA Systinet uses ports 8080 and 8443. If you select a different set of ports during installation, you must configure JBoss after installation to use these ports. If you are using port numbers that are higher than the default, the easiest way is to edit the JBoss configuration files:

To edit the JBoss port numbers:

- 1 Open `JBOSS_HOME/server/CONFIG_HOME/conf/jboss-service.xml` in an editor. (`CONFIG_HOME` refers to the JBoss configuration to which you have deployed SOA Systinet. For non-clustered deployments, this is usually `default` and for clustered deployments, this is usually `all`.)
- 2 Search for the string `ports-01`. The search function takes you to the following commented-out MBean:

```
<!-- (comment text).....
<mbean code="org.jboss.services.binding.ServiceBindingManager"
      name="jboss.system:service=ServiceBindingManager">
  <attribute name="ServerName">ports-01</attribute>
  <attribute name="StoreURL">
    ${jboss.home.url}/docs/examples/binding-manager/sample-bindings.xml
  </attribute>
  <attribute name="StoreFactoryClassName">
    org.jboss.services.binding.XMLServicesStoreFactory
  </attribute>
</mbean>
-->
```

- 3 Remove the wrapping comment tag and comment text from the MBean.
- 4 Set the value of the element `<attribute name="ServerName">ports-01</attribute>`. This value represents the factors of 100 by which additional port numbers above the default value are enabled. For example, if you leave the value at `ports-01`, ports 8180, 8280, 8380... are enabled. If you set the value at `ports-02`, the additional ports are 8280, 8480, 8680...
- 5 Save your changes and exit the editor.

Deploying SOA Systinet to a JBoss Cluster

SOA Systinet installers do not support clustered installation. It is therefore necessary to install to a temporary standalone JBoss configuration and then manually create cluster nodes from that standalone JBoss.

This section gives instructions for creating a cluster JBoss application servers where each hosts one JBoss instance with SOA Systinet. A JBoss instance with a configuration named `node1` is located on the first machine, `node2` on the second one, and additional nodes as required. A load balancer runs on the first machine.

For details, see the following sections:

- [Load Balancing on JBoss on page 66](#)
- [Installing SOA Systinet to a JBoss Cluster on page 69](#)

Load Balancing on JBoss

The following instructions are for the use of the `mod_jk` module in Apache 2.2 but you can use any passive-cookie load balancer which is supported by JBoss. For more information about `mod_jk`, see [the Apache documentation](#) [http://tomcat.apache.org/tomcat-3.3-doc/mod_jk-howto.html]. You can download `mod_jk` from [the Apache site](#) [<http://tomcat.apache.org/connectors-doc/>]. There is also a version you can copy and paste in [Example 2 on page 67](#).

Example 2. Pasteable mod_jk.conf

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk-apache-2.2.3.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
JkMount /* loadbalancer

# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkermmap.properties

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
    JkMount status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
```

</Location>

To set up `mod_jk` load balancing:

- 1 Install an Apache server, or configure an existing Apache server, to use the ports and host name which will be used for SOA Systinet. Also configure SSL if it is required for deployment.
- 2 Copy `mod_jk.conf` to `APACHE/conf`.
- 3 In the Apache Tomcat `/conf` directory, edit `httpd.conf`. Add the line `Include conf/mod_jk.conf` to the end of the file. Make other changes to `httpd.conf` as described in that file's comments and in the Apache documentation.
- 4 Modify contexts in the file `APACHE/conf/uriworkermap.properties`, if necessary.
- 5 Modify workers settings in the file `APACHE/conf/workers.properties`. Change `worker.nodeName.port`, `worker.nodeName.host`, `worker.loadbalancer.balance_workers` and the number of workers. Names of nodes (`nodeName`) must match names of corresponding JBoss configurations. [Example 3 on page 69](#) is a modified `workers.properties` file.
- 6 Run the Apache server with the configured load balancer.

Example 3. Modified workers.properties

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=server1
worker.node1.type=ajpl3
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=server2
worker.node2.type=ajpl3
worker.node2.lbfactor=1

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1

# Status worker for managing load balancer
worker.status.type=status
```

Installing SOA Systinet to a JBoss Cluster

Install SOA Systinet to a temporary JBoss configuration.

To install SOA Systinet when clustering JBoss:

- 1 Copy the `JBOSS_HOME/server/all` configuration. Name the copy `nodeX`.
- 2 Launch the SOA Systinet installer.
- 3 In the **Endpoint Properties** step, pass the load balancer hostname and endpoints.

- 4 In the **Application Server Properties** step, deploy to the `nodeX` configuration.

When the load balancer is running and SOA Systinet is installed to a temporary configuration, create the first node to use for running SOA Systinet.

To create the first node:

- 1 Copy the `JBOSS_HOME/server/all` configuration. Name the copy `node1`.
- 2 Set up JMS for your database.

For details, see [Configuring JMS for JBoss on page 54](#).

- 3 Copy the `nodeX` datasource, `JBOSS_HOME\server\nodeX\deploy\hpssoasystinet-xa-ds.xml`. Paste it into `JBOSS_HOME\server\node1\deploy\`
- 4 Copy the `nodeX` mail configuration, `JBOSS_HOME\server\nodeX\deploy\mail-service.xml`. Paste it into `JBOSS_HOME\server\node1\deploy\`
- 5 Enable the use of the `mod_jk` load balancer. Set the value of the `UseJK` attribute to `true` in the file `JBOSS_HOME\server\node1\deploy\jboss-web.deployer\META-INF\jboss-service.xml`
- 6 Open the file `JBOSS_HOME\server\node1\deploy\jboss-web.deployer\server.xml` for editing.
- 7 Comment out the HTTP connector listening at port 8080.

This step is optional, but an existing HTTP listener can hide a misconfiguration or a bug.

- 8 Add the attribute `jvmRoute="{jboss.server.name}"` to the `Engine` element with the name `jboss.web`. (Do not evaluate the attribute value. Place it in the configuration file as is. It will be evaluated by JBoss at runtime.) The `jvmRoute="{jboss.server.name}"` attribute appends a suffix with the node name to outgoing `JSESSIONID` headers. These suffixes are used by the load balancer to maintain session affinity.
- 9 Apply the following workaround to disable session replication among clusters:
 - a Open the file `JBOSS_HOME\server\node1\deploy\jboss-web-cluster.sar\META-INF\jboss-service.xml` for editing.
 - b Change the value of `buddyReplicationEnabled` from `false` to `true`.

Change the value of `numBuddies` from 1 to 0.

- 10 Copy the following files from `JBOSS_HOME\server\nodeX\conf` to `JBOSS_HOME\server\node1\conf`:
 - `roles.properties`
 - `users.properties`
 - `server.cer`
 - `server.keystore`
- 11 Create an additional cluster node.
 - a Copy your JBoss installation to a second computer.
 - b Create a `JBOSS_HOME/server/node2` directory on that computer.
 - c Copy the directory `JBOSS_HOME\server\node1` to the `node2` directory on the second computer.
 - d Repeat for `node3`, `node4`...`nodeN`.
- 12 Copy the following files from `JBOSS_HOME\server\nodeX\deploy` to `JBOSS_HOME\server\node1\farm\`. They are distributed to all other cluster nodes when those nodes boot.
 - `hp-soa-systinet.ear`
 - `hp-soa-systinet.sar`
- 13 Launch `node1` on the first computer. When it successfully starts, launch `node2` node on the second computer. Continue for all other nodes. For each node, it is necessary to specify the URL of the HA-JNDI service in the local JBoss. Base the command-line for starting a node on the following:

```
JBOSS_HOME\bin\run.bat -b 0.0.0.0 -c nodeName  
-Dhpsoa.hajndi.url-jnp://hostname:1100/  
-Djboss.partition.name=DefaultPartition
```

14 SOA Systinet should be running on `http://balancerHostname:port/soa/`

Setting Up Oracle Application Server

SOA Systinet requires some initial configuration of OAS before you can deploy to it.



In this document, `OAS_HOME` refers to the installation directory for the Oracle Application Server.

The procedures for setting up OAS for SOA Systinet are described in the following sections:

- [Update JDBC Driver on page 72](#)
- [Allocating Memory and Setting Java Options on page 73](#)
- [Creating Resources on page 75](#)
- [Setting Up SSL on OAS on page 80](#)

Update JDBC Driver

The version of Oracle Application Server that SOA Systinet supports has an out-of-date JDBC driver. Therefore, you must download and install a new copy of `ojdbc14dms.jar`.

To update the JDBC driver for OAS:

- 1 Make sure OC4J is not running.
- 2 Open a command console in `OAS_HOME/product/10.1.3.1/oas1/jdbc/lib`.
- 3 Rename the existing `ojdbc14dms.jar` file to `ojdbc14dms.jar.backup`.
- 4 In your web browser, open the [Oracle JDBC driver download page](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html) [http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html] and download version 10.2.0.4 or later of `ojdbc14dms.jar`.
- 5 Copy the newly downloaded `ojdbc14dms.jar` to `OAS_HOME/product/10.1.3.1/oas1/jdbc/lib`.

Allocating Memory and Setting Java Options

SOA Systinet requires the following changes to the Oracle Application Server memory and Java settings:

- Increase the memory maximum permanent size to 512MB.
- Increase the memory initial heap size to 800MB.
- Increase the memory maximum heap size to at least 1000MB, depending on your deployment.
- Set the Java property, `-Djava.awt.headless=true`.
- Set the Java property, `-Doc4j.userThreads=true`.

You can make these changes with the Oracle administration console or by editing `opmn.xml`.

To edit the settings with the console:

- 1 Start the Administration Console for the OC4J instance that will host SOA Systinet.
- 2 Open the Administration tab.
- 3 From the menu, select **Administration Tasks**→**Properties**→**Server Properties**→**Command Line Options**.
- 4 Set the Initial Heap Size to **800M**.
- 5 Set the Maximum Heap Size to at least **1000M**.
- 6 Add or reset the following Java options:
 - `-XX:MaxPermSize=512M`
 - `-Djava.awt.headless=true`
 - `-Doc4j.userThreads=true`
- 7 Click **Apply**.

8 Restart the OC4J instance.

To edit the settings in `opmn.xml`:

1 Make sure the OC4J instance is not running.

2 Open `OAS_HOME/opmn/conf/opmn.xml` with a text editor.

3 Modify or add the following settings to the start parameters:

- `-XX:MaxPermSize=512M`
- `-Xms800M`
- `-Xmx1000M`
- `-Djava.awt.headless=true`
- `-Doc4j.userThreads=true`



Memory sizing should take performance requirements into consideration for the deployed system. These settings are only a recommendation.

4 Save your changes.

5 Restart the OC4J instance.

Example 4. opmn.xml Example

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<opmn xmlns="http://www.oracle.com/ias-instance">
  ...
  <process-manager>
    ...
    <ias-instance id="server1.domain.com" name="server1.domain.com">
      ...
      <ias-component id="default_group">
        <process-type id="home" module-id="OC4J" status="enabled">
          <module-data>
            <category id="start-parameters">
              <data id="java-options" value="-Xrs -server -XX:MaxPermSize=512M -Xms800M -Xmx1000M
                -XX:AppendRatio=3 -Djava.security.policy=$ORACLE_HOME/j2ee/home/config/java2.policy
                -Djava.awt.headless=true -Dhttp.webdir.enable=false -Doc4j.userThreads=true"/>
            </category>
            ...
          </module-data>
          ...
        </process-type>
      </ias-component>
    </ias-instance>
  </process-manager>
</opmn>
```

Creating Resources

SOA Systinet requires JDBC and JMS resources to communicate with the database.

In OAS, use the Application Server Control to create them.

To use the Application Server Control to create resources:

- 1 Start OAS with the following command:
OAS_HOME/opmn/bin/opmnctl startall
- 2 Open the Application Server Control (<http://localhost/em>).
- 3 Log in as the OC4J administrator.

- 4 Click the link to your OC4J instance (by default named Home) in the tree view on the main page.
- 5 Open the Administration tab.

Use the Administration tab to create JDBC and JMS resources.

For details, see the following sections:

- [Creating JDBC Resources on page 76](#)
- [Creating JMS Resources on page 78](#)

Creating JDBC Resources

SOA Systinet requires a global JDBC datasource to communicate with the database. Before creating this datasource, you must create a connection pool.



SOA Systinet uses XA transactions. The application server transaction manager should be configured to have a minimum of 5 minutes for XA transaction timeout.

For details, see your application server documentation.

In the Administration tab, create a connection pool and then a datasource that uses the connection pool.

To create a connection pool:

- 1 From the menu, select **Administration Tasks**→**Services**→**JDBC Resources**.
- 2 In the Connection Pools section, click **Create**.
The Create Connection Pool - Application wizard opens.
- 3 Leave Application as **default**, select **New connection pool**, and then click **Continue**.
A details page for a new connection pool opens.
- 4 Name the connection pool [HP SOA Systinet Connection Pool](#).
- 5 Keep the default Connection Factory Class (`oracle.jdbc.pool.OracleDataSource`).

- 6 Select **Generate URL From Connection Configuration** and enter the following connection property fields:

Field name	Value
Driver Type	Thin
DB Host Name	<i>Type the database host name</i>
DB Listener Port	<i>Type the database port</i>
DB Identifier Type	SID
SID/Service Name	<i>Type sid</i>
Username (under Credentials section)	<i>Type your database username</i>
Use Cleartext Password / Password	<i>Type your database password</i>



Use the same database parameters for SOA Systinet installation.

- 7 Click **Test Connection**. This tests the connection properties you entered in [Step 6](#). Check the connection properties if the test fails.
- 8 Open the Attributes tab and enter the following parameters:

Field name	Value
Initial size of Connection Cache	5
Minimum Number of Connections	5
Maximum Number of Connections	200
Validate Connection	True

- 9 Click **Finish**.

After you create a connection pool, you can use it to create a datasource.

To create a datasource:

- 1 From the menu, select **Administration Tasks**→**Services**→**JDBC Resources**.
- 2 In the Data Sources section, click **Create**.

The Create Data Source - Application & Type wizard opens.

- 3 Leave **Application** as **default**, select **Managed data source**, and then click **Continue**.

A details page for a new datasource opens.

- 4 Enter the following parameters:

Parameter	Value
Name	HP SOA Systinet Data Source
JNDI Location	hpsoasystinetDS
Transaction Level	Global and Local Transactions
Connection Pool	HP SOA Systinet Connection Pool
Login Timeout	0 (<i>default</i>)

Leave the credentials fields empty. The connection pool credentials will be used.

- 5 Click **Finish**.

Creating JMS Resources

SOA Systinet requires connection factories and destinations.



You can configure JMS to meet your requirements. This section describes a JMS set up that ensures that JMS resources are accessible by SOA Systinet and function correctly.

To create JMS connection factories for SOA Systinet:

- 1 From the menu, select **Administration Tasks**→**Services**→**Enterprise Messaging Service**→**JMS Connection Factories**.

- 2 Click **Create New**.

A wizard opens for creating a connection factory.

- 3 Create the following connection factories. Each row corresponds to one connection factory. Leave blank all fields not included in the following table. After you enter the values for a connection factory, click **OK**.

Repeat from [Step 2](#) until you have created all the connection factories.

Connection Factory Type	JNDI Location	Check "XA Enabled"?
Queue	jms/ReportingSenderConnectionFactory	Yes
Queue	jms/ReportingReceiverConnectionFactory	Yes
Queue	jms/PMConnectionFactory	Yes
Queue	jms/platform/QueueConnectionFactory	Yes
Topic	jms/platform/TopicConnectionFactory	Yes

To create JMS destinations for SOA Systinet:

- 1 From the menu, select **Administration Tasks**→**Services**→**Enterprise Messaging Service**→**JMS Destinations**.

- 2 Click **Create New**.

A wizard opens for creating a JMS destination.

- 3 Create the following destinations. Each row corresponds to one destination. Leave blank all fields not included in the following table. After you enter the values for a destination, click **OK**.

Repeat from [Step 2](#) until you have created all the destinations.

Destination Type	Destination Name	Select Persistence	JNDI Location	Persistence File
Queue	RF Executions Queue	File Based	queue/ReportingExecutions	RF_ReportingExecutionsQueue
Queue	Platform scheduleTimerQueue	File Based	queue/scheduleTimerQueue	platform_scheduleTimerQueue
Queue	Platform taskProcessorQueue	File Based	queue/taskProcessorQueue	platform_taskProcessorQueue
Topic	Platform taskStopperTopic	File Based	topic/taskStopperTopic	platform_taskStopperTopic
Queue	PM Validation Queue	File Based	queue/Validation	PM_validationQueue
Queue	PM Priority Validation Queue	File Based	queue/PriorityValidation	PM_priorityValidationQueue

Setting Up SSL on OAS

The Oracle HTTP server contains a demo certificate. To use SSL communication for SOA Systinet, you must create a real certificate for OAS to trust.



This is an optional step, you can use the demo certificate if it meets the requirements for applications that connect to SOA Systinet.

To create a certificate for use by the Oracle HTTP Server:

- 1 Launch the Oracle Wallet Manager in one of the following ways:
 - On Windows, select *your_server_name* **Integrated Management Tools**→**Wallet Manager** from the GUI menu.
 - On UNIX systems, launch Wallet Manager with the `own` script in `OAS_HOME/bin`.
- 2 In the Wallet Manager GUI menu, select **Wallet**→**New**.

The Create New Wallet wizard opens.

- 3 If you are asked to create a default wallet directory, select **No**.
- 4 Select **Standard** wallet type.
- 5 Enter a password for the wallet.



This password must be alphanumeric, with at least 8 characters.

- 6 Click **OK**.

An empty wallet is created.

- 7 When prompted to create a certificate for the wallet, click **Yes**.

A detail page for the certificate opens.

- 8 For the certificate common name (CN), enter the hostname of the OAS server.



The hostname must be written exactly as it appears in the URL of the OAS server.

- 9 Enter other details, according to your company policy on creating certificates.

- 10 From the menu, select **Wallet**→**Save**.

When prompted, enter the following location:

`OAS_HOME\Apache\Apache\conf\ssl.wlt\new.`

Confirm that you want to create this directory.

- 11 From the menu, select **Wallet**→**Auto Login**.

- 12 From the menu, select **Wallet**→**Save**.
- 13 Export the certificate request in one of the following ways:
 - a From the menu, select **Operations**→**Export Certificate Request**.
 - b You are prompted for the type of certificate file and the location.
Select the CSR file type and the location of your choice.
- 14 Exit Wallet Manager.
- 15 Request a Certification Authority (CA) to sign your exported certificate.
It may be signed by one of the following authorities:
 - A public certification authority, such as VeriSign.
 - A corporate certification authority. Check your company IT security guidelines.
 - Yourself, with the OpenSSL tool.
- 16 When you obtain a signed copy of your certificate from the CA, you should also obtain a copy of the CA's own certificate.
- 17 Reopen Wallet Manager and the wallet you created.
- 18 From the menu, select **Operations**→**Import Trusted Certificate**.
Import the certificate of the CA that signed your certificate.
- 19 From the menu, select **Operations**→**Import User Certificate**.
Import the signed copy of your certificate.
- 20 From the menu, select **Wallet**→**Save**, and then exit Wallet Manager.
- 21 Set up Oracle HTTP server to use the wallet you created.

- Open `OAS_HOME\Apache\Apache\conf\ssl.conf` in an editor.
- Change the `SSLWallet` directive to point to the newly generated wallet file at `OAS_HOME\Apache\Apache\conf\ssl.wlt\new`.

Setting Up WebLogic

Set up a separate WebLogic domain to host SOA Systinet. You must configure the domain itself and JDBC and JMS properties for the managed servers and/or clusters in the domain. Review [Designing Your Deployment on page 15](#), you need to know the number and location of cluster servers or managed servers before you start.



Make sure you use the correct version of WebLogic as stated in [Supported Platforms on page 10](#).



In this document, `WL_HOME` refers to the WebLogic server installation directory.

WebLogic set up is described in the following sections:

- [Creating Domains for SOA Systinet on page 84](#)
- [Setting Up a WebLogic Managed Server on page 85](#)
- [Creating Resources on page 88](#)
- [Starting SOA Systinet in WebLogic on page 99](#)



The **Lock and Edit** and **Activate Changes** steps do not apply to WebLogic 10g (10.3) in Development mode as any changes made are directly applied. If you are using WebLogic 10g (10.3) in development mode, skip these steps.

Creating Domains for SOA Systinet

You must host SOA Systinet in a separate WebLogic domain.

To create a new WebLogic domain using the WebLogic Configuration Wizard:

- 1 Launch the WebLogic Configuration wizard with the following command:

WL_HOME/common/bin/config



In Windows you can launch the wizard from the **Start** menu.

- 2 Select **Create a new WebLogic Domain**, and click **Next**.
The Select Domain Source page opens.
- 3 You can use the default settings for the domain source and then click **Next**.
The Admin Username page opens.
- 4 Set the WebLogic administrator username and password, and click **Next**.
The Configure Server Start Mode and JDK page opens.
- 5 You can select either **Development** or **Production** mode.
- 6 Select the JDK according to [Supported Platforms on page 10](#) and then click **Next**.
The Customize Environment and Services Settings page opens.
- 7 Select **Yes** and click **Next**.
- 8 You can use the default Administrator Server settings. Click **Next**.
- 9 Create the managed servers required by your deployment (see [Designing Your Deployment on page 15](#)). You require at least one managed server. Do not install SOA Systinet to the administration server.

Give the servers arbitrary names, such as `hpsystinet1`. You can use a proxy for clusters. Make sure the server ports do not conflict with the administration server.

- 10 Create any clusters required for your deployment and then click **Next**.
- 11 If you use clusters, assign managed servers to them and then click **Next**.
- 12 Optionally, create a HTTP proxy for your clusters, and then click **Next**.
- 13 Create a machine in your domain and then click **Next**.
- 14 Assign all managed servers, both clustered and standalone, to the machine. You do not have to assign the administration server to the machine. Click **Next**.
- 15 Review the new domain and click **Next**.
- 16 Set the domain name and location and click **Create**.



In this document, the domain is referred to as `hpsoa_domain`. The location is referred to as `DOMAIN_HOME`.

Setting Up a WebLogic Managed Server

Each managed server hosting SOA Systinet requires some configuration.

To set up a managed server:

- 1 On each machine hosting the administration server, a managed server, or a cluster, start the WebLogic Node Manager with the following command:

```
WL_HOME/server/bin/startNodeManager
```



WebLogic 10g (10.3) node manager uses system variables `PATH` and `CLASSPATH` in the server start command. The node manager does not handle these variables if they contain spaces.

To avoid this problem, do one of the following:

- On Windows, replace the conflicting parts of the paths with DOS-like 8.3 file names and restart node manager.
- Edit `WL_HOME/common/nodemanager/nodemanager.properties`, and add the parameter `StartScriptEnabled=true`, and then restart node manager.

2 Start the WebLogic server for your domain with the following command:

DOMAIN_HOME/startWebLogic

3 In your browser, open the WebLogic Administration Console:

`http://localhost:7001/console`

4 Log in with the administrator credentials created in [Creating Domains for SOA Systinet on page 84](#).

5 In the web console in the Change Center section,, click **Lock & Edit**.

6 In the Domain Structure section, expand **Services** and select **JTA**.

The Domain Settings page opens.

7 HP Software recommend setting **Timeout Seconds** to 300 and clicking **Save**.

8 In the Change Center section, click **Activate Changes**.

9 In the Domain Structure section, expand **Environment** and select **Servers**.

The Summary of Servers page opens.

10 In the Change Center section, click **Lock & Edit**.

11 For each managed server set the start-up parameters, [Step 12](#) to [Step 15](#).

12 In the Summary of Servers page, click the new server name.

The Settings page opens.

13 Select the Configuration: Server Start tab.

14 Set the class path to the following:

- For WebLogic 9.2:

```
DB_DRIVER_PATHS;JAVA_HOME/lib/tools.jar;WL_HOME/server/lib/weblogic_sp.jar;WL_HOME/server/lib/weblogic.jar
```

- For WebLogic 10g (10.3):

```
DB_DRIVER_PATHS;JAVA_HOME/lib/tools.jar;WL_HOME/server/lib/weblogic_sp.jar;  
WL_HOME/server/lib/weblogic.jar;BEA_HOME/modules/features/weblogic.server.modules_10.3.0.0.jar;  
BEA_HOME/modules/features/com.bea.cie.common-plugin.launch_2.1.0.0.jar
```



DB_DRIVER_PATHS must contain the paths to all JARs for the drivers required by the database supported by SOA Systinet. For details, see [Supported Databases on page 14](#).

JAVA_HOME is the location of the JDK that WebLogic uses and BEA_HOME is set during WebLogic installation.

On UNIX systems, use colons instead of semi-colons as the file separator.

For Oracle Database, the classpath must include the file location for ora_i18n.jar

15 Add the following to Arguments:

- For JRockit:

```
-Xms1024m -Xmx1024m
```

- For Sun or HP JDK:

```
-Xms1024m -Xmx1024m -XX:MaxPermSize=512m
```



Memory sizing should take performance requirements into consideration for the deployed system. These settings are only a recommendation.



If you use a UNIX operating system, also add the following property:

Generic JVM Arguments `-Djava.awt.headless=true`

16 Click **Save** and **Activate Changes**.



If you start servers from the command line or a script, the Java arguments you add in the Administration Console are not applied. Start the managed server with commands based on these scripts:

startMyNode.bat for Windows:

```
set DOMAIN_HOME=c:\your\weblogic\domain\home
set JAVA_OPTIONS="-Djava.awt.headless=true"
set USER_MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=512m"
%DOMAIN_HOME%\bin\startManagedWebLogic [nodename]
```

startMyNode.sh for Linux:

```
export DOMAIN_HOME=/your/weblogic/domain/home
export JAVA_OPTIONS="-Djava.awt.headless=true"
export USER_MEM_ARGS="-Xms1024m -Xmx1024m -XX:MaxPermSize=512m"
. $DOMAIN_HOME/bin/startManagedWebLogic.sh [nodename] http://hostname:7001
```

The managed server is now available to start. HP Software recommend setting up resources, as described in [Creating Resources on page 88](#), before starting the managed server. The managed server must be running if you start SOA Systinet from the Deployments page of the WebLogic Administration console.

Creating Resources

SOA Systinet requires a number of resources to be set up in WebLogic.

Use the Administration Console to create them.

To use the Administration Console to create resources:

- 1 In your browser, open the WebLogic Administration Console:

`http://localhost:7001/console`

- 2 Log in as the WebLogic administrator.
- 3 Use the Administration Console to create mail, JDBC, and JMS resources.

For details, see the following sections:

- [Creating Mail Sessions on page 89](#)
- [Creating JDBC Resources on page 90](#)
- [Creating JMS Resources on page 95](#)
- [Setting Up the Security Realm on page 99](#)

You can verify your resources set up in your browser:

`http://localhost:7001/console/consolejndi.portal`

Creating Mail Sessions

SOA Systinet requires a mail session for automated notifications.

To create an SOA Systinet Mail Session:

- 1 In the Domain Structure section, expand **Services** and select **Mail Sessions**.
The Summary of Mail Sessions page opens.
- 2 Click **Lock & Edit**.
- 3 Click **New**.
The Create a New Mail Session page opens.
- 4 Enter a name, and then click **OK**.
- 5 In the Summary of Mail Sessions page, click the new mail session name.

- 6 Enter the JNDI Name, `/Mail`, enter JavaMail Properties according to your environment mail settings, for example, `mail.transport.protocol=smtp;mail.user=builder;mail.smtp.host=mail.com`), and then click **Save**.



JNDI names must be exact.

- 7 In the Settings page, select the **Targets** tab.
- 8 Target all servers and clusters hosting SOA Systinet, and click **Save**.
- 9 Click **Activate Changes**.

Creating JDBC Resources

SOA Systinet requires two JDBC datasources, an XA-enabled datasource and a non-XA-enabled datasource. These datasources handle all traffic between the SOA Systinet on WebLogic and the database server. Each WebLogic managed server and/or cluster server requires a persistent store on the database, which uses the non-XA-enabled datasource for communication.

Use the WebLogic Administration Console to create JDBC datasources. The Administration Server must be running.



SOA Systinet uses XA transactions. The application server transaction manager should be configured to have a minimum of 5 minutes for XA transaction timeout.

For details, see your application server documentation.

To create an XA-enabled JDBC datasource:

- 1 Open the Summary of JDBC Datasources page (Services→JDBC→Data Sources). Click **Lock and Edit** then **New**.
- 2 Give the datasource a unique, arbitrary descriptive name, such as HP SOA Systinet DS
- 3 Give the datasource the JNDI name `hpsoasystinetDS`.



JNDI names must be exact.

- 4 From the Database drop-down list, select the same database type that you use for SOA Systinet.
- 5 From the Database Driver drop-down list, select an XA-supporting JDBC database driver for the database type.



If you are using Oracle, select the Oracle "thin" XA driver.

- 6 Click **Next** to open the Transaction Options page and click Next again to open the Connection Properties page.
- 7 In the Connection Properties page, use the same database parameters you use for SOA Systinet. Proceed to Target Selection by clicking **Next**.
- 8 In Select Targets, select all servers or clusters hosting SOA Systinet.
- 9 Click **Finish** to return to the Summary of JDBC Datasources page, and click **Activate Changes**.

The datasource you created appears in the table of datasources.

- 10 Click **Lock and Edit**.
- 11 Click the name of the XA datasource in the table of datasources to open its details page in the Configuration:General tab.
- 12 Open the Configuration:Connection Pool tab.

Increase the maximum capacity of the connection pool. The Maximum Capacity should be at least 1/4 the number of parallel requests that you require to be handled simultaneously. If you do not have an estimate of this number, set the maximum capacity to 100.

- 13 Increase the Initial Capacity to the number of expected concurrent users.

- 14 Click **Save** but do not activate changes.
- 15 To enable failover, in the Connection Pool tab, expand the Advanced section.
Select **Test Connections on Reserve**.
- 16 Open the Configuration:Transactions tab ensure that **Use XA Datasource Interface** is selected.
- 17 Click **Save** but do not activate changes.
- 18 Navigate out of the datasource details page, for example to the Summary of JDBC Datasources page, and click **Activate Changes**.



If you do not navigate out of the datasource details page before you save changes to the datasource, you cause a JDBCSystemResourceMBean cannot be null exception. This exception is harmless, because the changes to the datasource are activated anyway, but avoidable.



If the exception "Could not get JDBC Connection; nested exception is java.sql.SQLException: Internal error: Cannot obtain XAConnection weblogic.common.resourcepool.ResourceDisabledException: Pool hpssoasystinetDS is disabled, cannot allocate resources to applications..." occurs in the in log file, you can:

- Increase the count of connections in the datasource.
- Increase the timeout for acquiring a connection.

Increase the value of Connection Reserve Timeout (Data Sources→hpssoasystinetDS→Connection Pool→Advanced). (Default is 10 seconds).

Setting 0 (infinite waiting for connection) is not recommended because of a risk of deadlocks.

To create a non-XA-enabled JDBC datasource:

- 1 Open the JDBC datasources page (Services→JDBC→Data Sources). Click **Lock and Edit** and then **New**.
- 2 Give the datasource a unique, arbitrary descriptive name such as SOA Systinet JMS DS.
- 3 Give the datasource the JNDI name jms-hpsoasystinetDS.



JNDI names must be exact.

- 4 From the Database drop-down list, select the same database type that you use for SOA Systinet.
- 5 From the Database Driver drop-down list, select a non-XA-supporting JDBC database driver for the database type.



If you are using Oracle, select the Oracle "thin" non-XA driver.

- 6 Click **Next** to open the Transaction Options page.
- 7 Select **Supports Global Transactions** and click **Next** to open the Connection Properties page.
- 8 In the Connection Properties page, use the same database parameters you use for SOA Systinet. Proceed to Target Selection by clicking **Next**.
- 9 In Select Targets, select all servers or clusters hosting SOA Systinet.
- 10 Click **Finish** and **Activate Changes**. The datasource you created appears in the table of datasources.
- 11 Click **Lock and Edit**.
- 12 To enable failover, click the name of the non-XA datasource in the table of datasources to open its details page in the Configuration:General tab.

- 13 Open the Configuration:Connection Pool tab and expand the Advanced section.

Select **Test Connections on Reserve**.

Click **Save** but do not activate changes.

- 14 Navigate out of the datasource details page, for example to the Summary of JDBC Datasources page, and click **Activate Changes**.



If you do not navigate out of the datasource details page before you save changes to the datasource, you cause a JDBC `SystemResourceMBean cannot be null` exception. This exception is harmless, because the changes to the datasource are activated anyway, but avoidable.

Create a JDBC persistent store for every migratable cluster server and every standalone server hosting SOA Systinet. These persistent stores use the `jms-hpsoasystinetDS non-XA` datasource.

To create a JDBC persistent store:

- 1 Navigate to Services→Persistent Stores.
- 2 Click **Lock and Edit**.
- 3 From the New drop-down menu, select **Create a JDBC Store** to open the Create JDBC Store wizard.
- 4 Give the persistent store a unique, arbitrary name, such as `SERVER_NAME` Store.
- 5 From the Target drop-down list, select the standalone managed server or migratable cluster server corresponding to the selected persistent store.
- 6 In the Datasource drop-down field, select the non-XA-enabled datasource you created.
- 7 Give the persistent store a unique prefix, so that the stores do not use the same table.
- 8 Click **Finish** and save your changes.
- 9 Repeat the procedure for each migratable cluster server and/or standalone managed server.
- 10 Click **Activate Changes**.

Creating JMS Resources

SOA Systinet requires JMS resources to be set up in WebLogic.



You can configure JMS to meet your requirements. This section describes a JMS set up that ensures that JMS resources are accessible by SOA Systinet and function correctly.

Create a JMS server for each migratable cluster server and each standalone server.

To create a JMS Server:

- 1 Open the JMS Servers page, Services→Messaging→JMS Servers.
- 2 Click **Lock and Edit** and then **New** to open the Create a New JMS Server wizard in the JMS Server Properties page.
- 3 In the Name field, give the JMS server a unique, arbitrary descriptive name, such as *SERVER_NAME* JMS, indicating which server is targeted.
- 4 From the Persistent Store drop-down field, select the persistent store of the managed server or migratable cluster server to target.

Click Next to open the Select Targets page

- 5 From the Target drop-down field, select the standalone managed server or migratable cluster server corresponding to the persistent store you selected.

Click **Finish** and **Activate Changes**.

Create a JMS Module to contain definitions of JMS connection factories as well as required JMS destinations.

To create a JMS module:

- 1 Open the JMS Modules page, Services→Messaging→JMS Modules.
- 2 Click **Lock and Edit** and **New** to open the Create JMS System Module wizard.
- 3 Give the JMS module a unique, arbitrary descriptive name, such as HP SOA Systinet JMS Module.

You may apply any descriptor file name and location, or leave those fields blank to use the default.

Click **Next**.

- 4 Target the standalone server or the cluster hosting SOA Systinet.
- 5 Select **Would you like to add resources to this JMS system module?**, and click **Finish**.

The details page for the JMS Module opens in the Configuration tab.

- 6 Create the connection factories listed in [Table 17, "JMS Connection Factories for WebLogic"](#).

Table 17. JMS Connection Factories for WebLogic

Name	JNDI Name
HP SOA Systinet Connection Factory	/ConnectionFactory
Reporting Sender Connection Factory	jms/ReportingSenderConnectionFactory
Reporting Receiver Connection Factory	jms/ReportingReceiverConnectionFactory

- a Click **New** in the Summary of Resources table to open the Create a New JMS System Module Resource wizard.
- b Select **Connection Factory** and click **Next** to open the Create a New Connection Factory wizard.
- c Give the connection factory a unique, arbitrary descriptive name (for example the one listed in [Table 17, "JMS Connection Factories for WebLogic"](#).)
- d Give the connection factory the JNDI name specified in the JNDI Name column.



JNDI names must be exact.

- e Use the default targeting, which selects the parent module target.
- f Click **Finish**.

- g Edit each connection factory and select **XA Connection Factory Enabled** in the Configuration:Transactions tab.
 - h If the SOA Systinet host is a cluster, open the Configuration: Load Balancing tab and disable server affinity.
 - i Navigate back to the list of JMS Modules and then click **Activate Changes**.
- 7 If the SOA Systinet host is a managed server and not a cluster create a subdeployment.
- a Click **Lock & Edit**.
 - b Navigate to the SOA Systinet JMS Module details page.
 - c Open the Subdeployments tab and create a subdeployment for the JMS module.
 - d Set the subdeployment target as the SOA Systinet host managed server's JMS server.
 - e Create the resources listed in the [Table 18, "JMS Resources for a WebLogic Managed Server"](#).

Return to the JMS module details page, and click **New** in the Resources table. Select the resource type and click **Next**. Leave blank all fields not included in the table. After you enter the values for a resource, configure it to use the subdeployment and click **Finish**.

Table 18. JMS Resources for a WebLogic Managed Server

Resource Type	Name	JNDI Name
Queue	SC scheduleTimerQueue	queue/scheduleTimerQueue
Queue	PM Validations Queue	queue/Validation
Queue	PM Priority Validations Queue	queue/PriorityValidation
Queue	SC TaskProcessorQueue	queue/taskProcessorQueue
Queue	RF Executions Queue	queue/ReportingExecutions
Topic	SC taskStopperTopic	topic/taskStopperTopic

- 8 If the SOA Systinet host is a cluster create the resources listed in the Table 19, "JMS Resources for a WebLogic Cluster".

Table 19. JMS Resources for a WebLogic Cluster

Resource Type	Name	JNDI Name
Distributed Queue	SC scheduleTimerQueue	queue/scheduleTimerQueue
Distributed Queue	PM Validations Queue	queue/Validation
Distributed Queue	PM Priority Validations Queue	queue/PriorityValidation
Distributed Queue	SC TaskProcessorQueue	queue/taskProcessorQueue
Distributed Queue	RF Executions Queue	queue/ReportingExecutions
Distributed Topic	SC taskStopperTopic	topic/taskStopperTopic

- a Click **Lock & Edit**.
 - b Navigate to the SOA Systinet JMS Module details page.
 - c Click **New** in the Resources table.
 - d Select the resource type and click **Next**.
 - e Input the Name and JNDI Name. Leave blank all fields not included in the table and then click **Next**
 - f After you enter the values for a resource, target your cluster and click **Finish**.
- 9 Click **Activate Changes**.

Setting Up the Security Realm

In SOA Systinet authentication is performed by the application server. You must set up the WebLogic Security Realm, otherwise you can only log in to SOA Systinet with the WebLogic administrator credentials.

In the Domain Structure section of the Administration Console navigate to Security Realms. Click **Configure New Security Realms** in the **How do I...** section to open the WebLogic guide to setting up a security realm or changing the default realm.

Create your realm according to your requirements for LDAP and WebLogic user store rights.



When using multiple Providers in your Security Realm, make sure that all providers have their Control Flag set to `SUFFICIENT`.

Any changes require a restart of the administration server and any managed servers.

Starting SOA Systinet in WebLogic

After installation, start SOA Systinet using WebLogic.

To start SOA Systinet in WebLogic:

- 1 Start a node manager:

```
WL_HOME/server/bin/startNodeManager
```

- 2 Start an administration server:

```
DOMAIN_HOME/startWebLogic
```

- 3 Start the server hosting SOA Systinet:

```
DOMAIN_HOME/bin/startManagedWebLogic.cmd MY_MANAGED_SERVER
```

Setting Up WebSphere

WebSphere requires initial configuration before you can deploy SOA Systinet to it.

WebSphere set up is described in the following sections:

- [Creating a Profile on page 100](#)
- [Creating a Mail Session on page 100](#)
- [Creating JDBC Resources on page 101](#)
- [Creating a Messaging Bus on page 106](#)
- [Setting Up JMS on page 108](#)
- [Setting Startup Parameters on page 112](#)
- [Setting Up a WebSphere Cluster on page 113](#)

Creating a Profile

Create a clean WebSphere profile with the *Cell* environment. This profile is stored in `WS_HOME/AppServer/profiles/PROFILE_NAME`.



In this document, the path is referred to as `PROFILE_HOME`.

Create application servers and/or clusters in the new profile as required by your deployment design. For details, see [Designing Your Deployment on page 15](#).

If you are using a web server such as IBM HTTP Server (IHS) as a proxy or load balancer, register it with the Deployment Manager. For details, see the WebSphere Help.

Creating a Mail Session

Create a mail session using the WebSphere Administration Console.

To create a mail session:

- 1 Open the WebSphere Administration Console:

`http://localhost:9060/ibm/console`

- 2 Select **Resources**→**Mail**→**Mail Sessions**.
- 3 Select your cell in the **Scope** drop-down field and click **New**.
- 4 Specify the mail session parameters as follows:
 - A unique, arbitrary descriptive name, for example `SOA Systinet Mail`
 - The JNDI name `/Mail`
 - Connection settings as per company email set up
 - SMTP credentials if required

Creating JDBC Resources

SOA Systinet requires an XA-enabled JDBC datasource to communicate with the database. JMS messaging requires a non-XA datasource.

Before creating these two datasources, you must create a JDBC provider for each of them.



SOA Systinet uses XA transactions. The application server transaction manager should be configured to have a minimum of 5 minutes for XA transaction timeout.

For details, see your application server documentation.

Open the WebSphere Administration Console (<http://localhost:9060/ibm/console>) and create JDBC resources, in the order of the following sections:

- 1 To create a JDBC provider for an XA datasource:
- 2 To create a JDBC provider for a non-XA datasource:
- 3 To create an XA-enabled JDBC data source:
- 4 To create a non-XA-enabled JDBC datasource (used by JMS):

To create a JDBC provider for an XA datasource:

- 1 Select **Resources**→**JDBC**→**JDBC Providers**.
- 2 For Scope, select your cell, and click **New**.
- 3 Select your database type.
- 4 Under Provider, select the driver for your database type.



For DB2, if there is more than one driver available, select the DB2 Universal driver.
For Oracle, if there is more than one driver available, select a "Thin" driver.

- 5 For the implementation type, select **XA data source**.

The **Name** is automatically completed with the driver name, followed by **(XA)**.

- 6 For the value of the variable `${driver_name_PATH}`, enter the location of the driver:



`ojdbc14.jar` (Oracle) or `db2jcc.jar` and `db2jcc_license_cu.jar` (DB2). The DB2 driver files are in `IBM_HOME/SQLLIB/java` by default.

- 7 Click **Finish**.

To create a JDBC provider for a non-XA datasource:

- Repeat [To create a JDBC provider for an XA datasource:](#), with the following exception:

Select the implementation type **Connection pool data source**.

The automatically generated name should not end in **(XA)**.



If you get the error DSRA3602E, refer to [Wadmin scripting fails with DSRA3602E](http://www-1.ibm.com/support/docview.wss?uid=swg21257580) [http://www-1.ibm.com/support/docview.wss?uid=swg21257580].

To create an XA-enabled JDBC data source:

- 1 Select **Resources**→**JDBC**→**Data Sources**.
- 2 For Scope, select your cell, and then click **New**.
- 3 Give the data source a unique, arbitrary descriptive name, for example, HP SOA Systinet DS.
- 4 Give the datasource the JNDI name `hpsoasystinetDS`.
- 5 Click **Next**.

The Select JDBC Provider page opens.

- 6 Select **Select an existing JDBC provider**, select the XA JDBC provider you created in [To create a JDBC provider for an XA datasource](#), and then click **Next**.

The Database Properties page opens.

- 7 The database properties you enter depend on the type of database you are using:

For DB2:

- Enter the database name, such as `platform`. Your database administrator can tell you this name.
- From the **Driver type** drop-down field, select driver type "4."
- Enter the server name.
- Enter the port number if it differs from the default 50000.
- Deselect **Use this datasource for CMP**.

For Oracle:

- Type the full URL of the database you plan to use for SOA Systinet, such as
jdbc:oracle:thin:@server:1521:database
- From the drop-down field, select the data store helper class name for your version of the database.
- Deselect **Use this datasource for CMP**.

8 Click **Finish**.

9 Open the newly created data source and create an authentication alias:

- Click **JAAS - J2C Authentication Data**.

A list of authentication aliases opens.

- Click **New**.
- Give an arbitrary string value for the alias, for example, HP SOA Systinet Credentials.
- For credentials, enter the user name and password for the database you use with SOA Systinet.
- Click **Finish**.

10 Reopen the newly created datasource and apply the new authentication alias:

- In Component-Managed Authentication Alias, select the authentication alias you created in [Step 9](#).
- Under Authentication Alias for XA Recovery, select **Use Component-Manager Authentication Alias**.
- Under Container-Managed Authentication, for the Mapping Configuration Alias, select **DefaultPrincipalMapping**.
- Click **OK**.

11 Reopen the datasource and increase its connection pool size:

- Under Additional Properties, click **Connection pool properties**.

The Connection Pool page opens.

- For Maximum Connections, type a number equal to at least 1/4 of the number of parallel requests that you require to be handled simultaneously. If you do not have an estimate of this number, set the maximum connections to 100.
- For Minimum Connections, type a number equal to the number of expected concurrent users.
- Click **OK**.

12 Click **Test connection** to make sure that your datasource configuration is correct.



If you get the error "Connection not available, Timed out waiting for 180000" in the log file, you can do the following:

- Increase the Maximum connections property in **Resources**→**JDBC**→**Datasources**→**HP SOA Systinet DS for Oracle**→**Connection pool properties** (or specify 0 for no connection count limit)
- Increase the value Connection Timeout property in **Resources**→**JDBC**→**Datasources**→**HP SOA Systinet DS on Oracle**→**Connection pool properties** (default is 180 seconds). Setting 0 (infinite waiting for connection) is not recommended because of a risk of deadlocks.

To create a non-XA-enabled JDBC datasource (used by JMS):

- Repeat [To create an XA-enabled JDBC data source:](#), with the following exceptions:
 - Give the non-XA datasource the JNDI name, `jms-hpsoasystinetDS`.
 - Select the non-XA JDBC provider.
 - Use the same authentication alias you created for the XA-enabled datasource.

After creating JDBC resources, restart the WebSphere Deployment Manager.

Creating a Messaging Bus

In WebSphere, JMS communication and the persistent storage of that communication are handled via a bus.

To create a messaging bus:

- 1 Open the WebSphere Administration Console:

`http://localhost:9060/ibm/console`

- 2 Select **Service Integration**→**Buses**.

The Buses page opens.

- 3 Click **New**.

The Create a New Messaging Engine Bus wizard opens.

- 4 Give the bus a unique, arbitrary descriptive name, for example, `SOABus`.

- 5 Deselect **Bus Security**, as it is not required, and click **Next**.

- 6 Click **Finish**.

The Buses page reopens.

- 7 Click the name of the bus you created to open its details page.

- 8 Click **Bus members**.

The Bus members page opens.

- 9 Click **Add**.

The Add a New Bus Member wizard opens.

- 10 Select a standalone server or cluster that will host SOA Systinet, and click **Next**.

The Select Type of Message Store page opens.

11 Select **Data store** for the type of message store, and click **Next**.

12 Enter the message store properties for the bus.



You can use the existing `jms-hpsoasystinetDS` data source but you might prefer to use a different data source for performance reasons.

If you use an existing data source, for the Schema Name, type the database user name, set the Authentication Alias to `HP SOA Systinet Credentials`, select **Create Tables**, and then click **Next**.

13 Review your selected options and click **Finish**.

The Bus Members page reopens.

14 Repeat [Step 9](#) to [Step 13](#) for every standalone server and cluster that will host SOA Systinet.

15 Return to the bus details page.

The Configuration tab is open by default.

16 Under Destination Resources, click **Destinations**.

A table of destinations opens.

17 Add the following destinations by clicking **New**:

Destination type	Identifier
Queue	scheduleTimerQueue
Queue	RFReportingExecutions
Queue	Validation
Queue	Priority Validation
Queue	taskProcessorQueue
Topic Space	taskStopperTopic

Setting Up JMS

SOA Systinet requires JMS messaging resources that you must set up in the WebSphere Administration Console.



You can configure JMS to meet your requirements. This section describes a JMS set up that ensures that JMS resources are accessible by SOA Systinet and function correctly.

To set up JMS:

- 1 Open the WebSphere Administration Console:

`http://localhost:9060/ibm/console`

- 2 Select **Resources**→**JMS**.
- 3 Add the JMS resources listed in [Table 20, "JMS Resources"](#).

For each resource:

- Under JMS, click the resource type.
- Select the scope created in [Creating a Profile on page 100](#), and click **New**.
- Select **Default Messaging Provider**, and click **OK**.
- Use the parameters from [Table 20, "JMS Resources"](#).

Use the bus you created in [Creating a Messaging Bus on page 106](#).

Where a Queue Name, or Topic Space is required, select the relevant queue or topic that you created in [Creating a Messaging Bus on page 106 Step 17](#).

- Click **OK**.

Table 20. JMS Resources

Resource Type	Name	JNDI Name
Queue Connection Factory	RF Connection Factory (Send)	jms/ReportingSenderConnectionFactory
Queue Connection Factory	RF Connection Factory (Rec)	jms/ReportingReceiverConnectionFactory
Queue Connection Factory	SOA Queue Connection Factory	jms/SOAQueueConnectionFactory
Topic Connection Factory	SOA Topic Connection Factory	jms/SOATopicConnectionFactory
Queue	SC scheduleTimer Queue	queue/scheduleTimerQueue
Queue	PM Validations Queue	queue/Validation
Queue	PM Priority Validations Queue	queue/PriorityValidation
Queue	SC TaskProcessorQueue	queue/taskProcessorQueue
Queue	RF Executions Queue	queue/ReportingExecutions
Topic	SC taskStopperTopic	topic/taskStopperTopic

- 4 Modify the connection pool for the SOA Queue and the SOA Topic connection factories, by doing the following:
 - Select the connection factory to modify.
 - Under Additional Properties, click **Connection Pool Properties**.
 - Change Maximum Connections to 100.
 - Click **OK**.

- 5 Select **Resources**→**Asynchronous Beans**, and select **Work Managers**.

6 Select the scope you created in [Creating a Profile on page 100](#), and click **New** to create the following work managers:

- **SC Work Manager**

Set JNDI Name `/wm/platform`, change Maximum Number of threads to `100`, and select **Growable**.



The leading forward slash is required for the JNDI name in this case.

- **RF Work Manager**

Set JNDI Name `wm/reporting`.

7 Set the JMS Activation Specifications listed in [Table 21, "JMS Resources"](#).

- Under JMS, click the **Activation Specifications**.
- Select the scope created in [Creating a Profile on page 100](#), and then click **New**.
- Select **Default Messaging Provider**, and then click **Next**.
- Use the parameters from [Table 21, "JMS Resources"](#).

Use the appropriate Destination Type `Queue` or `Topic` for each resource.

Use the bus you created in [Creating a Messaging Bus on page 106](#).

- Click **OK**.

Table 21. JMS Resources

Name	JNDI Name	Destination JNDI Name
RF Activation	jms/RFActivation	queue/ReportingExecutions
PM Activation	jms/PMActivation	queue/Validation

Name	JNDI Name	Destination JNDI Name
PM Priority Activation	jms/PMPriorityActivation	queue/PriorityValidation
PL Scheduler Timer Queue Activation	jms/PLSchedulerTimerQueueActivation	queue/scheduleTimerQueue
PL Task Runner Queue Activation	jms/PLTaskRunnerQueueActivation	queue/taskProcessorQueue
PL Task Stopper Topic Activation	jms/PLTaskStopperTopicActivation	topic/taskStopperTopic

- 8 Expand **Service Integration**, and select **Buses**.
- 9 Select the bus you created in [Creating a Messaging Bus on page 106](#).
- 10 In the Topology section, click **Messaging Engines**.
- 11 Copy the name of the messaging engine to the clipboard.
- 12 Select **Resources**→**JMS**, and select **Activation Specifications**.
- 13 "Maximum Concurrent Endpoints" must be decreased to 3 on "PM Activation" in JMS - Activation specifications.

"Maximum Concurrent Endpoints" must be decreased to 2 on "PM Priority Activation" in JMS - Activation specifications.
- 14 Select **PL Task Stopper Topic Activation**.
- 15 In the Subscription Durability section, add the following parameters:
 - Leave Subscription Durability as non-durable.
 - Enter a subscription name, for example, "PL Task Stopper Subscription Name"
 - Enter a client identifier, for example, "PL Task Subscription ID"

- Paste the messaging engine name as the Durable Subscription Home.

16 Click **OK**.

Setting Startup Parameters

SOA Systinet requires several parameters to be set in order to function correctly.

Use the WebSphere Administration Console to set these startup parameters.

To set the startup parameters:

1 In your browser, open the WebSphere Administration Console:

`http://localhost:9060/ibm/console`

2 Expand **Servers**, and select **Application Servers**.

3 Select the server.

4 In the Server Infrastructure section, expand **Java and Process Management**, and select **Process Definition**.

5 In the Additional Properties section, select **Java Virtual Machine**.

6 Set the following properties:

- Initial Heap Size `1000`
- Maximum Heap Size `1500`
- Generic JVM Arguments `-XX:MaxPermSize=150m`



Memory sizing should take performance requirements into consideration for the deployed system. These settings are only a recommendation.



If you use a UNIX operating system, also add the following property:

Generic JVM Arguments `Djava.awt.headless=true`

7 Click **OK**.



You must restart the server for these changes to take effect.

Setting Up a WebSphere Cluster

Clustered deployment of SOA Systinet is very similar to standalone deployment.

In all the following set up procedures, make sure to do the following:

- Whenever you select deployment scope, choose the cluster itself.
- When a restart is necessary, restart the whole cluster, including all servers joined to the cluster.
- When configuring a cluster, configure all servers within the cluster.
- When you deploy SOA Systinet, map modules to servers by selecting the cluster and an instance of *IBM HTTP Server*.

The following procedure describes how to set up a proxied load balanced cluster with two servers running on one node.

To create a load balanced cluster:

- 1 Install and start *IBM HTTP Server*.
- 2 Create a new WebSphere cell (deployment manager and application server) profile, by doing the following:
 - Start the WebSphere Profile Management Tool.

- Select **Cell**, and click **Next**.
- Select **Advanced profile creation**.
- Select **Deploy the administrative console** (recommended), and **Deploy the default application**.
- Enter a unique Deployment Manager Profile Name (DMGR_NAME), Application Server Profile Name (APPSRV_NAME), and select a location for the new profile, and then click **Next**.
- Enter the Deployment Manager Node Name and the Application Server Node Name.



These become the node containing the clustered servers.

If necessary, correct the Host Name and Cell Name (CELL_NAME), and then click **Next**.

- Select **Enable administrative security**, enter the administrator credentials, and then click **Next**.
- If required, change the port values, and click **Next**.
- Deselect **Run the deployment manager process as a Windows service**, and click **Next**.
- Click **Next**, and then **Create**.
- Deselect **Launch the First Steps Console**, and click **Finish**.

3 Start the deployment manager, application server node, and the application server:

- a Execute the command:

```
PROFILE_HOME/DMGR_HOME/bin/startManager
```

- b Execute the command:

```
PROFILE_HOME/APPSRV_HOME/bin/startNode
```



If you want to run one or more nodes of the cluster on different machines, follow this procedure for each machine.

To apply a cluster to other machines:

- 1 Install IBM WebSphere®.
- 2 Start the WebSphere Profile Management Tool.
- 3 Click **Next**, and select **Custom Profile**.
- 4 Select **Advanced Profile Creation**.
- 5 Enter the Profile Name, for example, HPsoaClusterAppsrv2, and Profile Directory (PROFILE2_HOME).
- 6 Select **Make this profile default** and click **Next**.
- 7 Enter the Node Name (for example, HpsoaClusterNode2), and click **Next**.
- 8 Enter the Deployment Manager Hostname or IP Address, pointing to an existing deployment manager in a cell you want the new node to federate with.

Set the credentials to administer the new deployment manager, optionally change the deployment manager port, and then click **Next**.
- 9 If necessary, change the port values, and click **Next**.
- 10 Start the node with the following command:

PROFILE2_HOME/bin/startNode

The new node should appear in the deployment manager admin console nodes listing, **System Administration**→**Nodes**.

- 4 To create the cluster:

- a In your browser, open the WebSphere Administration Console:

`http://localhost:9060/admin`



The port may vary depending on your settings.

- b Select **Servers**, and select **Clusters**.
- c Click **New**.
- d Enter a cluster name (CLUSTER_NAME), select **Configure HTTP session memory-to-memory replication**, and then click **Next**.
- e In the Select Basis for First Cluster Member, select **Create the member by converting an existing application server**, and click **Next**.
- f Enter a new member name, for example `server2`, and click **Add Member**.
- g Add servers as required.
If you require a different node, select it from **Select Node**.
- h Click **Next**, and **Finish**, and then **Save**.
- i During the resources set up for SOA Systinet, described in the previous sections, use the CELL_NAME as the target for the resource.
- j During SOA Systinet deployment, select the CELL_NAME as the target for deployment.

- 5 To add a web server node to the cluster:

- a In your browser, open the WebSphere Administration Console:

`http://localhost:9060/admin`



The port may vary depending on your settings.

- b Select **Servers**→**Web Servers**, and click **New**.
 - c Enter a Server Name, for example, `IHS_NAME`, and click **Next** twice.
 - d Enter the Web Server Location, which should be the installation directory for *IBM HTTP Server* (`IHS_HOME`), and the a Plug-in Installation Location, usually, `IHS_HOME/Plugins`.
 - e Click **Next**, and **Finish**, and then **Save**.
- 6 For debug purposes, add an alias to the default virtual host to enable direct access to applications on all clustered servers, by doing the following:
- a In your browser, open the WebSphere Administration Console:

```
http://localhost:9060/admin
```



The port may vary depending on your settings.

- b Select **Environment**, and select **Virtual Hosts**.
 - c Click **default_host**, and then click **Host Aliases**.
 - d Click **New**.
 - e Enter the value for the second clustered server port, usually `9081`, click **OK**, and then click **Save**.
Repeat this step for as many servers as you require, in addition to adding their ports.
- 7 Regenerate the routing information for the web server.

Select **Servers**→**Web Servers**, and select `IHS_NAME`, and then select **Generate Plug-in** and **Propagate Plug-in**.

8 To start the cluster:

- In your browser, open the WebSphere Administration Console:

```
http://localhost:9060/admin
```



The port may vary depending on your settings.

- Select **Servers**, and then **Clusters**.
- Click your `CLUSTER_NAME`.
- Click **Start**.
- When the server starts, you can validate if the requests are load balanced across the cluster.

In your browser, use the URL:

```
http://localhost/snoop
```

The Snoop servlet page should appear, displaying the port number.

Every time you refresh the page, the port number should change.

```
http://localhost/soa/web
```

 should display the SOA Systinet user interface.

LDAP Accounts Integration

When you install SOA Systinet, you can select to use an external LDAP server to retrieve information about users and groups.

The following sections describe how to integrate accounts from an LDAP server into SOA Systinet:

- [Automatic Service Discovery on page 119](#)

A brief explanation of automatic service discovery and its implications.

- [LDAP Service Properties on page 119](#)

A list of JNDI properties of the LDAP server that must be known to SOA Systinet.

Automatic Service Discovery

The automatic discovery of LDAP servers means you do not have to hardwire the URL and port of the LDAP server. Instead you can use `ldap:///o=JNDITutorial,dc=example,dc=com` as a URL, and the real URL is deduced from the distinguished name `o=JNDITutorial,dc=example,dc=com`.

Automatic discovery of the LDAP service using the URL's distinguished name is supported only in Java 2 SDK, versions 1.4.1 and later, so make sure that your Java version supports this.

LDAP Service Properties

To integrate external accounts during SOA Systinet installation, do one of the following:

- In the GUI Installer, select **LDAP** in the User Management page, and set your LDAP properties.

For details, see [LDAP Options on page 32](#).

- Set the LDAP properties in the `configuration-properties.xml` file during manual deployment.

For details, see [Configuring the User Store on page 127](#).

SOA Systinet integration with LDAP uses a JNDI interface to connect to LDAP servers.

For more information, about the JNDI API, see <http://java.sun.com/products/jndi/tutorial/ldap/connect/create.-html> and <http://java.sun.com/j2se/1.5.0/docs/guide/jndi/jndi-dns.html#URL> .

The following JNDI properties must be known to the server:

Property Name	Property Description	API Link
Naming Provider URL	URL of the LDAP service.	http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/Context.html#PROVIDER_URL
Initial Naming Factory	Java class for the initial naming factory.	http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/Context.html#INITIAL_CONTEXT_FACTORY
Security Principal	The name of the security principal for anonymous read access to the directory service.	http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/Context.html#SECURITY_PRINCIPAL
Password	Password of security principal.	http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/Context.html#SECURITY_CREDENTIALS
Security Protocol	Name of the security protocol. Default is "simple."	http://java.sun.com/j2se/1.5.0/docs/api/javax/naming/Context.html#SECURITY_PROTOCOL

Siteminder Integration

You can configure SOA Systinet to use Siteminder as an authentication proxy.

To integrate SOA Systinet with Siteminder

- 1 In Siteminder, configure SOA Systinet endpoint authentication. For details of default SOA Systinet authentication, see "Default Endpoint Authentication" in the *HP SOA Systinet Developer Guide*.
- 2 During installation, set the SOA Systinet endpoint to a proxy server integrated with Siteminder.
For details, see [Step 18 of Using the GUI Installer on page 19](#) or [Configuring the Endpoint on page 133](#).
- 3 After installation, use the Setup Tool to configure the use of cookies or headers.

For details, see "Setting Up Siteminder Integration" in the *HP SOA Systinet Administrator Guide*.



WebDAV functionality is unavailable for SOA Systinet integrated with Siteminder because Siteminder does not support the WebDAV protocol.

4 Deploying SOA Systinet

The GUI Installation Wizard described in [Chapter 2, Basic Installation](#), can be used for production environments. However, HP Software recommend using command-line installation for production deployment as the configuration options can be much more complicated.

Command line options are available with the installation command.

For details, see [Installation Command Line Options](#) on page 124.

Command-line deployment consists of the following steps:

- 1 Extract the installation archive.

For details, see [Unpacking the Distribution](#) on page 126.

- 2 Set the configuration properties for your deployment.

For details, see [Configuring the Deployment](#) on page 126.



For JBoss with HP-UX, set property `shared.as.jboss.preloading.classes.at.startup` to `true`.

Installation with some DB2 JDBC drivers do not manage explicit SQL commit commands in auto-commit mode. If commit errors occur during installation with DB2, set property `shared.installer.db.exclude.commits` to `true`.

- 3 Prepare any optional additions to your deployment.

For details, see the following sections:

- [Preparing Extensions](#) on page 139
- [Preparing Updates](#) on page 140

4 Finish the installation in one of the following ways:

- With decoupled database deployment.

For details, see [Decoupled Database Deployment on page 140](#).

- Including database deployment.

For details, see [Finishing Installation on page 142](#).

5 For applications servers other than JBoss, you must deploy the EAR file created by the installer.

For details, see [Deploying the EAR File on page 143](#).

[Deploying to Environments without a JDK on page 154](#) describes how to deploy SOA Systinet to environment running with a JRE but without a JDK.



Throughout this chapter, `SOA_HOME` refers to the SOA Systinet installation.

Installation Command Line Options

As an alternative to using the GUI installer, there are a number of command line options available:

The installation command is:

```
java -jar hp-soa-systinet-3.20.jar [OPTIONS]
```



If you have multiple JDKs installed, the first JDK in the path is used.

To ensure that the correct JDK is used:

- 1 Open a command prompt (cmd in Windows) or a terminal session (UNIX/Linux).
- 2 Execute **echo %JAVA_HOME%** (Windows) or **echo \$JAVA_HOME** (UNIX/Linux)
- 3 Do one of the following:

- If JAVA_HOME points to JDK 1.5 then proceed with installation.
 - If JAVA_HOME does not point to JDK 1.5 then reset the JAVA_HOME environment variable to a valid JDK 1.5.
- 4 Verify that JAVA_HOME is set correctly.
 - 5 Execute "%JAVA_HOME%/bin/java" -jar hp-soa-systinet-3.20.jar (Windows) or "\$JAVA_HOME/bin/java" -jar hp-soa-systinet-3.20.jar (UNIX/Linux).

The available option, with their definitions, are as follows:

- **-h, --help**

Display the available options or list the available scenarios or steps in the console.

- **-x, --extract *PATH***

Extract the installation archive to the specified location.

- **-i, --install-to *SOA_HOME***

Install SOA Systinet in console mode to the specified location. Normally used in conjunction with **-u**.

- **-s, --save-config *FILE***

Execute the GUI Installation, but save the configuration to the specified file instead of installing SOA Systinet.

- **-a, --dbadmin-mode**

Run the installation in decoupled database mode.

For details, see [Decoupled Database Deployment on page 140](#).

- **-u, --use-config *FILE***

Use the properties in the specified XML file to override the default or current configuration properties.

- **--passphrase** *PASSPHRASE*

If you want to use password encryption, specify the passphrase to use for encryption.

- **-d, --debug**

Execute the installation in debug mode. All properties, SQL statements, and installation details are output to `SOA_HOME/log/install.log`.

Unpacking the Distribution

SOA Systinet is distributed as a JAR file. To perform manual configuration, you must extract the archive first.

To unpack the JAR file:

- Execute the following command:

```
java -jar hp-soa-systinet-3.20.jar -x SOA_HOME
```

Configuring the Deployment

The distribution contains a configuration properties file. You can configure all your deployment options in this file prior to deployment.

To edit the configuration file:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 In the configuration file, make the changes you need for your deployment.

For configuration details, see the following sections:

- [Configuring the License on page 127](#)
- [Configuring the User Store on page 127](#)
- [Configuring the Administrator and System Mail Account on page 131](#)

- [Configuring the Database on page 132](#)
- [Configuring the Endpoint on page 133](#)
- [Configuring an Application Server on page 136](#)
- [Configuring Password Encryption on page 137](#)
- [Configuring Custom Data Import on page 138](#)
- [Configuring an SMTP Server on page 138](#)
- [Configuring Full Text Search on page 138](#)

3 Save your changes.

Configuring the License

If you have a license from your HP sales representative then you can use it in the configuration file.

To configure the license:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `shared.license.licensed.to` to the license owner.
- 3 Set `shared.license.key` to your license key.
- 4 Save your changes.

Configuring the User Store

During command-line installation, you can set the user store to be in the database or an LDAP server.

To configure the user store:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Do one of the following:

- Set `shared.um.account.backend.type=database` to use the database.
- Set `shared.um.account.backend.type=ldap` to use an LDAP server.

3 By default, SOA Systinet logins are case-insensitive. To enable case-sensitive logins, modify the following property:

```
shared.um.account.caseInsensitiveLoginName=false
```

4 Save your changes.

To configure LDAP integration:

1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.

2 Set the following properties according to your LDAP configuration:

- `shared.um.java.naming.provider.url`
- `shared.um.java.factory.initial=com.sun.jndi.ldap.LdapCtxFactory`
- `shared.um.java.naming.security.principal`
- `shared.um.java.naming.security.credentials`
- `shared.um.java.naming.security.authentication=simple`

3 To enable LDAP account integration, set the following properties:

Mandatory account properties:

- `shared.um.account.backend.type=ldap`
- `shared.um.account.backend.enableMoreBackends=true`
- `shared.um.account.ldapLoginName`
- `shared.um.account.ldapFullName`

- `shared.um.uddi.ldap.searchfilter.user`
- `shared.um.uddi.ldap.searchbase.user`

Optional account properties:

- `shared.um.uddi.ldap.searchscope.user=2`
- `shared.um.uddi.ldap.searchMaxResults.user=100`
- `shared.um.account.ldapEmail`
- `shared.um.account.ldapDescription`
- `shared.um.account.ldapLanguageCode`
- `shared.um.account.ldapBusinessName`
- `shared.um.account.ldapPhone`
- `shared.um.account.ldapAlternatePhone`
- `shared.um.account.ldapAddress`
- `shared.um.account.ldapCity`
- `shared.um.account.ldapCountry`
- `shared.um.account.ldapZip`

4 To enable LDAP group integration, set the following properties:

Mandatory group properties:

- `shared.um.group.backend.type=ldap`
- `shared.um.group.backend.enableMoreBackends=true`
- `shared.um.account.ldapMember`

- `shared.um.account.ldapName`

Optional group properties:

- `shared.um.uddi.ldap.searchfilter.group=2`
- `shared.um.uddi.ldap.searchbase.group=100`
- `shared.um.group.ldapDescription`
- `shared.um.account.ldapOwner`

5 To enable LDAP connection pooling, set the following properties:

- `shared.um.ldap.connect.pool=true`
- `shared.um.ldap.connect.pool.authentication`

A list of space-separated authentication types of connections that may be pooled. Valid types are `none`, `simple`, and `DIGEST-MD5`.

- `shared.um.ldap.connect.pool.debug`

A string that indicates the level of debug output to produce. Valid values are `fine` (trace connection creation and removal) and `all`.

- `shared.um.ldap.connect.pool.initsize`

The string representation of an integer that represents the number of connections per connection identity to create when initially creating a connection for the identity.

- `shared.um.ldap.connect.pool.maxsize`

The string representation of an integer that represents the maximum number of connections per connection identity that can be maintained concurrently.

- `shared.um.ldap.connect.pool.protocol`

A list of space-separated protocol types of connections that may be pooled. Valid types are `plain` and `ssl`.

- `shared.um.ldap.connect.pool.timeout`

The string representation of an integer that represents the number of milliseconds that an idle connection may remain in the pool without being closed and removed from the pool.

For more details about connection pooling, see <http://java.sun.com/products/jndi/tutorial/ldap/connect/config.html>.

If any of these connection pool properties are not set, they use the default values as described in the above link.

6 Save your changes.



Ensure that your mappings are correct and that these properties exist on your LDAP server. The incorrect mapping of any properties, even optional ones, can have a severe performance impact for sign-in for some LDAP services.



It may be more convenient to integrate LDAP accounts after installation using the Setup Tool.

For details, see "Setting Up LDAP Integration" in the *HP SOA Systinet Administrator Guide*.

Configuring the Administrator and System Mail Account

SOA Systinet requires an initial administrator user and a system mail account.

To configure the administrator and system mail:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `shared.administrator.username` and `shared.administrator.password` to the credentials for your administrator.



The administrator login name must be valid for the selected application server instance. The user with the specified name becomes an SOA Systinet administrator. For JBoss the specified administrator account is automatically created.

- 3 Set `shared.administrator.mail` to the email for the administrator.
- 4 Set `shared.notification.default.mail.from` to the mail address to be the source of system messages.
- 5 Save your changes.

Configuring the Database

SOA Systinet must connect to a correctly configured database.

For details about setting up a database, see [Preparing Databases on page 40](#).

To configure the database:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `shared.db.driver.path` to the location of your JDBC driver, including the driver file name.
 - For Oracle Database, use `ojdbc14.jar`. If you use localization you may also require `orail8n.jar`.
- 3 Set `shared.db.type` to one of the following options:
 - `db2`
 - `mssql`
 - `oracle`
- 4 Set `shared.db.username` and `shared.db.password` to the values provided by your database administrator.
- 5 Set `shared.db.host`, `shared.db.port`, and `shared.db.dbname` to the values for your database.

Alternatively, set `shared.db.url` to the connection string for your database.

- 6 Set `shared.db.action` to one of the following options:
 - To create a new schema in an existing database: `createSchema`
 - To create a new database/tablespace: `create`

For more details, see [Database Installation Types on page 41](#).

- 7 If you are creating a tablespace, set `shared.db.datafile` to the location of the tablespace, and set `shared.db.admin.username` and `shared.db.admin.password` to the database administrator credentials.
- 8 If you are using DB2, set `shared.db.tablespace` and `shared.db.pool` to the values set in [Setting Up IBM DB2 on page 46](#).
- 9 Save your changes.

Configuring the Endpoint

SOA Systinet is visible to the user as a web application and requires an endpoint.

You can configure the endpoint for a number of scenarios which should meet any connection requirement.



The default port numbers for each application server are different. Ensure that the port numbers you select match those required by your application server.

To configure the endpoint:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Do one of the following:
 - For HTTP only communication:
 - Set `shared.hostname` to the required server, for example, `localhost`.
 - Set `shared.http.port` to the required port, for example, `8080`.

Do not set `shared.https.port`. No port is required for this scenario.

Set `shared.https.use=false`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- For HTTPS only communication:

Set `shared.hostname` to the required server, for example `localhost`.

Do not set `shared.http.port`. No port is required for this scenario.

Set `shared.https.port` to the required secure port, for example, `8443`.

Set `shared.https.use=false`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- For HTTP and HTTPS with enforcing secured URLs:

Set `shared.hostname` to the required server, for example, `localhost`.

Set `shared.http.port` to the required port, for example, `8080`.

Set `shared.https.port` to the required secure port, for example, `8443`.

Set `shared.https.use=true`.

Set `shared.application.context` to your required deployment context., for example, `soa`.

- For HTTP and HTTPS without enforcing secured URLs:

Set `shared.hostname` to the required server, for example, `localhost`.

Set `shared.http.port` to the required port, for example, `8080`.

Set `shared.https.port` to the required secure port, for example, `8443`.

Set `shared.https.use=false`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- For an HTTP only proxied endpoint:

Set `shared.hostname` to the required proxy server, for example, `proxyhost`.

Set `shared.http.port` to the required port, for example, `8080`.

Do not set `shared.https.port`. No port is required for this scenario.

Set `shared.https.use=false`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- For an HTTPS only proxied endpoint:

Set `shared.hostname` to the required proxy server, for example, `proxyhost`.

Do not set `shared.http.port`. No port is required for this scenario.

Set `shared.https.port` to the required secure port, for example, `8443`.

Set `shared.https.use=false`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- For HTTP and HTTPS proxied endpoints with enforcing secured URLs:

Set `shared.hostname` to the required proxy server, for example, `proxyhost`.

Set `shared.http.port` to the required port, for example, `8080`.

Set `shared.https.port` to the required secure port, for example, `8443`.

Set `shared.https.use=true`.

Set `shared.application.context` to your required deployment context, for example, `soa`.

- 3 If your HTTPS communication is validated against SSL certificates, set `shared.ui.link.authentication=true`.

For details about SSL Certificates, see [SSL Certificates on page 176](#).

- 4 Save your changes.



If you configure SOA Systinet to run behind a load balancer proxy of with Siteminder you may need to configure the Setup Tool to ignore running servers on startup.

For details, see "Setup Tool" in the *HP SOA Systinet Administrator Guide*.

Configuring an Application Server

You must deploy SOA Systinet to an application server which is correctly set up.

For details about setting up an application server, [Setting Up Application Servers on page 52](#).

To configure an application server:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.

- 2 To set the application server settings:

For JBoss:

- Set `shared.as.server=jboss`
- Set `shared.as.version` to one of the following values:
 - For JBoss 4.2.2, use `422`.
 - For JBoss 4.3.0, use `430`.
- Set `shared.as.jboss.configuration` to the required configuration name.

By default, this is `default` for both JBoss 4.2.2 and JBoss 4.3.0.

- Set `shared.as.jboss.location` to the location of your application server installation folder.

For Oracle Application Server:

- Set `shared.as.server=oas`

For WebLogic:

- Set `shared.as.server=weblogic`

For WebSphere:

- Set `shared.as.server=websphere`

- 3 Save your changes.

Configuring Password Encryption

By default, the installer does not encrypt passwords in the configuration file.

If you require passwords to be encrypted for communication with other servers, such as the database and SMTP, you can enable encryption.

To enable password encryption:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `password.encryption.enabled=true`.
- 3 Save your changes.
- 4 During the next installation step, add **--passphrase** *PASSPHRASE* to the setup command to set the passphrase.

When you continue the installation, you must apply a passphrase to encrypt any passwords in the configuration file.

For details, see [Finishing Installation on page 142](#).



After installation with encryption, for the setup tool and some other command line tools, you must provide the passphrase protecting the encryption using the **--passphrase** command line option.

Configuring Custom Data Import

By default, the installer imports a default data set. You can configure the installation to import a custom data set instead.



The data set must conform to the requirements of the import tool.

For details, see "Import Tool" in the *HP SOA System Administrator Guide*.

To configure custom data import:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `import.type=custom`.
- 3 Set `import.path` to the location of your data image.
- 4 Save your changes.

Configuring an SMTP Server

By default, the installer does not configure an SMTP server for mail notifications.

To configure an SMTP server:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `platform.smtp.auth=true`
- 3 Set `platform.smtp.host` and `platform.smtp.port` to the location of your SMTP server.
- 4 Set `platform.smtp.auth.user` and `platform.smtp.auth.password` to an authorised SMTP server user.
- 5 Save your changes.

Configuring Full Text Search

By default, full text search is disabled in the configuration.

To enable full text search:

- 1 Open `SOA_HOME/conf/setup/configuration-properties.xml` with a text editor.
- 2 Set `shared.db.fulltextsearch=true`.
- 3 If required, to disable the automatic addition of % to search terms, set `shared.db.fulltextsearch.appendpercentage=false`.
- 4 Save your changes.

After installation you must enable full text search for your database, and in the SOA Systinet UI.

- Configure your database for full text search.

For details, see [Enabling Full Text Search on page 168](#).

- Enable full text search in the SOA Systinet UI.

For details, see "Configuration Options" in the *HP SOA Systinet Administrator Guide*.

Preparing Extensions

Most production deployments involve a customization layer with organization specific modifications to the SOA Definition Model (SDM), UI customization, and customized reports. These extensions can be added to the installation directory prior to deployment.

To add extensions prior to installation:

- 1 Extract the distribution file.

For details, see [Unpacking the Distribution on page 126](#)
- 2 Copy the extensions to the `SOA_HOME/extensions` folder.

Preparing Updates

To add updates prior to installation:

- 1 Extract the distribution file.
For details, see [Unpacking the Distribution on page 126](#).
- 2 Copy the required updates to the `SOA_HOME/updates` folder.

Decoupled Database Deployment

In some production deployments, the database administrator may perform the creation of the tablespace and schema as a separate installation step.

For more details, see [Manual Database Arrangement on page 42](#).

The installer enables you to separate the database deployment by creating scripts for the database administrator to run.

To perform decoupled database installation:

- 1 Extract the installation.
For details, see [Unpacking the Distribution on page 126](#).
- 2 Configure the installation properties.
For details, see [Configuring the Deployment on page 126](#).
For database details, see [Configuring the Database on page 132](#).



In this scenario you only require the database type, `shared.db.type`, and the required action, `shared.db.action`, for the database properties.

- 3 Execute the following command:

SOA_HOME/bin/setup -c -a



Add **--passphrase** *PASSPHRASE* if you want to use password encryption. For details, see [Configuring Password Encryption on page 137](#).

The installation stops after the creation of scripts, to create the tablespace and schema, depending on the configuration properties.

- 4 Provide the scripts created by the installer to the database administrator.

The installer creates the scripts in `SOA_HOME/sql`.

- If you are creating a new database/tablespace use the database administrator account to execute `createdb.sql`.
- To create the schema use the power user account to execute `all.sql`.



The schema creation scripts contain drop instructions which can, by design, fail and their failure must be ignored. If you are overwriting an existing SOA Systinet 3.20 database, make sure that the SQL tool ignores these failures.

For Oracle Database, `all.sql` executes a series of separate scripts to create the schema.



For more details about database arrangement, see [Database Installation Types on page 41](#).

The database administrator must execute the scripts and create a *common user* to provide SOA Systinet users access to the database.

For details, see the relevant common user procedure for each database in [Preparing Databases on page 40](#).

- 5 After the database administrator executes the scripts, set the database connection parameters in `configuration-properties.xml` as advised by the database administrator.

For details, see [Configuring the Database on page 132](#).

- 6 Execute the following command to finish the installation:

```
SOA_HOME/bin/setup -c
```



Add **--passphrase** *PASSPHRASE* if you want to use password encryption. For details, see [Configuring Password Encryption on page 137](#).

For more details about **setup -c**, see [Finishing Installation on page 142](#).

Finishing Installation

The installation command, **java -jar hp-soa-systinet-3.20.jar**, can only be executed once as the installation directory must be empty.

Command-line installation requires you to extract the installation archive first, using the **-x** switch on the installation command.

This extracts the file structure and the `configuration-properties.xml` file, enabling you to configure and prepare your deployment.

The Setup Tool enables you to continue installation after extraction.

To finish installation and deployment, do one of the following:

- Execute the command:

```
SOA_HOME/bin/setup -c
```

- If you enabled password encryption in [Configuring Password Encryption on page 137](#), execute the command:

```
SOA_HOME/bin/setup -c --passphrase PASSPHRASE
```

The installer uses `PASSPHRASE` to encrypt the password in `configuration-properties.xml`. For details, see [Configuring Password Encryption on page 137](#).

If you use a command line tool that requires authentication in another server, you must add the option `-passphrase PASSPHRASE` to the command.

The Setup Tool is a GUI utility which enables administrators to reconfigure a deployment of SOA Systinet. The `-c` switch runs the Setup Tool in console mode, continuing the installation from the point at which it previously stopped.

For details about the Setup Tool, see "Setup Tool" in the *HP SOA Systinet Administrator Guide*.

Deploying the EAR File

For non-JBoss application servers, you must deploy the EAR file created by the installer using application server functionality.

Additionally, you may need to modify the EAR file to apply authentication requirements.



If you are redeploying an EAR file, do not use the redeploy functionality of your application server. Undeploy the EAR file and then deploy the EAR file as described in this section.

If you are deploying to a cluster, you can only have one cluster server running. After a successful deployment, the deployment is copied to other cluster servers when you start them.



The SOA Systinet EAR file is updated during installation or by the application of extensions and updates. It contains JSPs that are compiled by the application server during deployment. The compilation of JSPs may take some time to complete.

HP Software recommend that you precompile JSPs before deployment, specifically for Oracle Application Server and WebLogic.

Use the following script to create an EAR file with precompiled JSPs.

```
SOA_HOME/deploy/AS/jspc/precompile
```

as is an application server specific folder name (jboss, oas, wl, or ws). The script may require some environment variables to be set. If they are not set, the script fails and outputs the name of the required environment variable.

The script creates `SOA_HOME/deploy/precompiled.ear` which can be used instead of `SOA_HOME/deploy/hp-soa-systinet.ear` during deployment.

For details, see the following sections:

- [Setting-up Authentication on page 144](#)
- [Role Mapping on page 146](#)
- [Deploying the EAR to OAS on page 146](#)
- [Deploying the EAR to WebLogic on page 150](#)
- [Deploying the EAR to WebSphere on page 151](#)

Setting-up Authentication

By default, SOA Systinet requires authentication for selected web resources. The configuration of these requirements conforms to the J2EE specification, as part of the deployment descriptors contained in the SOA Systinet EAR file.

[Table 22, "Authentication Methods"](#) describes the default authentication method with the URL patterns, relative to the deployment context of the EAR file (the default is `soa`).

Table 22. Authentication Methods

Authentication Method	URL Patterns
Form Authentication (required by the web UI)	web/service-catalog/* (Service Catalog UI)
	web/policy-manager/* (Policy Manager UI)
	web/shared/* (shared UI)
Basic Authentication (HTTP) (required by parts of the	systinet/platform/restBasic/* (see "Proprietary REST Interface" in the <i>HP SOA Systinet Developer Guide</i>)

REST interface and self-tester)	platform/restSecure/* (see "Atom-Based REST Interface" in the <i>HP SOA Systinet Developer Guide</i>)
	polycmgr/restSecure/* (Policy Manager REST interface)
	reporting/restSecure/* (Reporting REST interface)
	self-test/secure-snoop (see SOA Systinet Self-Test on page 37)
Authentication Method	URL Patterns
No authentication	web/design/* (static UI resources such as images)
	systinet/platform/rest/* (see "Proprietary REST Interface" in the <i>HP SOA Systinet Developer Guide</i>)
	platform/rest/* (see "Atom-Based REST Interface" in the <i>HP SOA Systinet Developer Guide</i>)
	polycmgr/rest/* (Policy Manager REST interface)
	reporting/rest/* (Reporting REST interface)
	self-test (excluding secure-snoop page, see SOA Systinet Self-Test on page 37)

The SOA Systinet EAR contains various WAR files. Some of the presented web pages may include links between resources contained in different WAR files. The security context (knowledge of the authenticated user) may be lost when following such links, so you may be prompted to sign in again.

Application servers provide a single-sign-on (SSO) solution for this situation:

- **JBoss**

SSO is set up during SOA Systinet installation.

For details, see <http://www.jboss.org/wiki/Wiki.jsp?page=SingleSignOn> .



If you setup 2-Way SSL with JBoss you must delete `WEB/INF/context.xml` in the `web-ui-war.war` file.

- **Oracle Application Server**

SSO must be enabled for the newly developed application.

For details, see http://download.oracle.com/docs/cd/B31017_01/web.1013/b28957/javasso.-htm#BABIDDAC.

- **WebLogic**

SSO is already set up in the deployment descriptor in the deployed EAR file.

- **WebSphere**

The SSO option is switched on when you enable administrative security.

Role Mapping

SOA Systinet requires one J2EE role, `authenticated`. By default, this role is mapped to any authenticated user for all application servers. If required, you can change the mapping of this role to grant or deny access for selected users that pass authentication.

For details, see the relevant security documentation for your application server.

SOA Systinet also contains an `administrator` role, which enables privileged access to all SOA Systinet resources independently of ACLs, as well as access to SOA Systinet administration tasks.

This role is managed by SOA Systinet and not by the application server. The initial administrator name is set during installation of SOA Systinet. Any administrator can use the SOA Systinet UI to assign the administrator role to additional users or user groups.

Deploying the EAR to OAS

The SOA Systinet installer does not deploy the EAR file to *Oracle Application Server (OAS)*, you must deploy it using OAS functionality.



If you are *re-deploying* an SOA Systinet EAR to OAS, *undeploy* the existing application and restart OC4J before deploying the updated EAR file.

To deploy the SOA Systinet EAR to OAS:

- 1 Setup your OAS server, specifically JDBC and JMS resources.

For details, see [Setting Up Oracle Application Server on page 72](#).

- 2 Do one of the following:

- Run the GUI installation using OAS parameters.

For details, see [Chapter 2, Basic Installation](#).

- Run command-line installation using OAS parameters.

For details about command line installation, see [Chapter 4, Deploying SOA Systinet](#).

- 3 Do one of the following:

- For OAS 10.1.3.4, execute the following deployment utility command:

```
OAS_HOME/j2ee/home/java -jar admin_client.jar OAS_URI OAS_ADMIN_UN OAS_ADMIN_PW -deploy -file SOA_HOME/deploy/hp-soa-systinet.ear -deploymentName "HP SOA Systinet" -bindAllWebApps
```

- *OAS_URI* is your OAS instance URI (for example `deployer:oc4j:localhost:12401`).
- *OAS_ADMIN_UN* is the OAS administrator username, usually `oc4jadmin`.
- *OAS_ADMIN_PW* is the OAS administrator password.

For details about command-line deployment go to the following location:

http://download-uk.oracle.com/docs/cd/B31017_01/web.1013/b28950/adminclient.htm#BABFHBGH

- For OAS 10.1.3.3 and lower, use the OAS GUI:

- 1 Start OAS with the command:

```
OAS_HOME/opmn/bin/opmnctl startall
```

- 2 Open the Application Server Control:

`http://localhost/em.`

- 3 Log in as the OC4J administrator.
- 4 Click the link to your OC4J instance (by default named Home) in the tree view on the main page.
- 5 Open the Applications tab.
- 6 Click **Deploy**.

The deployment wizard opens.

- 7 Do one of the following:

- Select **Archive is present on local host**, and **Browse** or enter the path to the EAR file.
- Select **Archive is already present**, and enter the path to the EAR file.



The default location of the EAR file is `SOA_HOME/deploy/hp-soa-systinet.ear`.

- 8 Keep the option **Automatically create a new deployment plan**, and click **Next**.

- 9 Enter the application name, `HP SOA Systinet`.

- 10 Leave all other fields at default, and click **Next**.

A page with the deployment details opens.

- 11 SOA Systinet relies on OAS to perform authentication. Your company policy determines the choice of a security provider for authentication.

The default security provider is usually file-based. Users and groups in a default file-based provider can be created using the OAS admin functions, but changes to users or groups may require an OC4J server restart to become effective.

You can also select an LDAP-based security provider.

For more details, see the *Oracle Containers for J2EE Security Guide*.

If you need to change the default security provider:

- In the Select Security Provider row, click **Go to Task**.
- Select the security provider you want. For example, for integration with Sun ONE Directory Server or Active Directory, select **Third Party LDAP Server**.
- Specify the provider details, and click **OK**.

12 Review the deployment details, and if they are correct click **Deploy**.



Deployment may take several minutes.

4 To verify successful deployment, in your browser, view:

```
http://hostname:port/context/self-test
```

A report page, HP SOA Systinet Self-test, is displayed and there should be no errors.

For self-test details, see [SOA Systinet Self-Test on page 37](#).

- 5 If you use a default file based security provider, you can manage users using the Oracle Application Server Control:
- a Open your OC4J instance.
 - b Select the **Administration** tab.
 - c Open **Security**→**Security Providers**.
 - d Click **Instance Level Security**.

- e Select the **Realms** tab.
- f Use **Create** to create the users you require including the admin user specified during installation.



Roles are not required.

Deploying the EAR to WebLogic

The SOA Systinet installer does not deploy the EAR file to WebLogic, you must deploy it using WebLogic functionality.

To deploy the SOA Systinet EAR to WebLogic:

- 1 Set up your WebLogic server.

For details, see [Setting Up WebLogic on page 83](#).

- 2 Do one of the following:

- Run command-line installation using WebLogic parameters.

For details, see [Chapter 4, Deploying SOA Systinet](#).

- Run the GUI installation using WebLogic parameters.

For details, see [Chapter 2, Basic Installation](#).

- 3 Start the WebLogic server.

- 4 In your browser, open the WebLogic Administration Console:

`http://localhost:7001/console`

- 5 Log in with the administrator credentials created in [Creating Domains for SOA Systinet on page 84](#).

- 6 In the console, click **Lock & Edit**.
- 7 In the Domain Structure section, select **Deployments**, and click **Install**.
- 8 Navigate to `SOA_HOME/deploy/`, select `hp-soa-systinet.ear`, and then click **Next**.
- 9 Select **Install this deployment as an application**, and click **Next**.
- 10 Select the managed server or cluster you want to host SOA Systinet, and click **Next**.
- 11 In the Security section, select **DD Only**, and click **Finish**.
- 12 Click **Activate Changes**.
- 13 Start the server hosting SOA Systinet:

```
DOMAIN_HOME/bin/startManagedWebLogic.cmd MY_MANAGED_SERVER
```

Start SOA Systinet using the **Start** on the Deployments page.

To verify that the SOA Systinet deployment is running, view self-test in a browser window, at `http://hostname:port/context/self-test`.

Deploying the EAR to WebSphere

The SOA Systinet installer does not deploy the EAR file to WebSphere, you must deploy it using WebSphere functionality.

To deploy the SOA Systinet EAR to WebSphere:

- 1 Set up your WebSphere server.
For details, see [Setting Up WebSphere on page 99](#).
- 2 Do one of the following:
 - Run command-line installation using WebSphere parameters.
For details, see [Chapter 4, Deploying SOA Systinet](#).

- Run the GUI installation using WebSphere parameters.

For details, see [Chapter 2, Basic Installation](#).

3 In your browser, open the Administration Console:

```
http://localhost:9060/ibm/console
```

4 Expand **Applications**, and select **Enterprise Applications**.

5 Click **Install**.

6 Click **Browse**, and navigate to `SOA_HOME/deploy/`, and then select `hp-soa-systinet.ear`.

7 Select **Prompt me only when additional information is required**, and click **Next**.

8 Set the following options:

- Precompile JavaServer Pages files
- Distribute application
- Allow dispatching includes to remote resources
- Allow servicing includes from remote resources

9 Map modules to servers by selecting the servers to deploy SOA Systinet.

10 Map modules to servers by selecting a module and virtual host.

11 Proceed to the Summary step, and click **Finish**.

12 Wait for the deployment to finish, and click **Save**.

13 Expand **Applications**, and select **Enterprise Applications**.

14 Select **HP SOA Systinet**.

15 In the Detail Properties section, click **Class loading and update detection**.

- 16 Set the following properties:
 - Polling interval 0
 - Classes loaded with application class loader first
 - Single class loader for application
- 17 Click **OK**, and save your changes.
- 18 Expand **Security**, and select **Secure administration, applications, and infrastructure**.
- 19 Select **Enable application security**. If the **Enable application security** option is disabled, then select **Enable administrative security** and **Enable application security**.

Deselect **Use Java 2 security to restrict access to local resources**.
- 20 In the Authentication section, expand **Web Security**, and select **General Settings**
- 21 Select **Use available authentication data when an unprotected URI is accessed** and click **OK**.
- 22 In the Configuration page, click **Apply**.
- 23 You can set users and roles if required.

To create a user:

- 1 Expand **Users and Groups**, and select **Manage Users**.
 - 2 Click **Create**.
 - 3 Enter the user parameters, and click **Create**.
-
- 24 Expand **Applications**, and select **Enterprise Applications**.
 - 25 Select the check-box for **HP SOA Systinet**, and click **Start**.



SOA Systinet starts automatically, whenever the server is started.



The HP SOA Systinet log can be viewed in the file:

```
PROFILE_HOME/logs/server_name/SystemOut.log.
```

Deploying to Environments without a JDK

This section describes a procedure for deploying SOA Systinet to production environments that cannot use a JDK, but only a JRE. The SOA Systinet installation framework supports this type of scenario.

In order to achieve this, two deployments are necessary, a staging environment called `Build` and a production environment called `Target`. The user responsible for installation is required to apply updates and extensions, and to compile JSPs on the `Build` machine that must use a JDK. Once the `Build` machine customization is complete, the results are transferred to the `Target` deployment.

This scenario requires a `Build` machine as the staging environment and a `Target` Deployment as the production environment.

The `Build` environment should mimic the `Target` deployment as much as possible:

- Install the application server and SOA Systinet to the same folders as required for the `Target`.
- Install the same version of the JDK as the JRE version on the target deployment. `JAVA_HOME` can differ from the `Target` deployment environment variable.
- The `Build` machine must use the same OS family as the `Target` deployment. This is required to generate compatible start scripts.



This process has been tested with JBoss only.

Installation (Build Machine)

- 1 Install the application server.
 - For JBoss, unzip the application server distribution.
 - For WebLogic domain, enter the full hostname (including domain) and port numbers for the target environment.
 - When possible (JBoss and WebLogic), it is useful to install the application server to the same folder where you intend to install SOA Systinet.

- 2 Generate the installation configuration file:

```
java -jar hp-soa-systinet-3.20.jar -s deployment.properties
```

- 3 Enter the full hostname (including domain) and port numbers for the Target environment in the properties file.

- 4 Delete the SOA Systinet extraction folder and execute the installation command:

```
java -jar hp-soa-systinet-3.20.jar -u deployment.properties -i /opt/hp/soa/systinet/3.20
```

- 5 Archive the clean deployment:

```
tar -cjf hp-soa-systinet-3.20-clean.tar.bz2 /opt/hp/soa/systinet
```



If possible, use `tar.gz` or `tar.bz2` to preserve executable flags.

Apply Updates (Build Machine)

- 1 Install any required SOA Systinet updates.
- 2 Apply any required extensions.
- 3 Import any required certificates.

- 4 Execute JSP script compilation:

```
./3.20/deploy/jboss/jspc/precompile_jsps.sh
```

- 5 Copy the precompiled EAR file to the application server and rename it `hp-soa-systinet.ear`.

- 6 Archive the SOA Systinet installation folder (including JBoss or WebLogic domain).

```
tar -cjf hp-soa-systinet-3.20-deployment-01.tar.bz2 /opt/hp/soa/systinet
```

Deployment (Target Environment)

- 1 Extract the archive:

```
tar -jxvf hp-soa-systinet-3.20-deployment-01.tar.bz2 /opt/hp/soa/systinet
```

- 2 Start the application server.



It is useful to keep previous versions of archived deployments, alongside exported data images to speed-up the process of updating or restoring a deployment.

5 After Installation

SOA Systinet may require some configuration after installation.

If you are upgrading from earlier versions of SOA Systinet you can migrate your data after installing SOA Systinet 3.20.

For details, see "Data Migration" in the *HP SOA Systinet Administrator Guide*.

This chapter contains the following sections:

- [Launching SOA Systinet on JBoss on page 157](#)
- [Enabling SSO in JBoss on page 158](#)
- [Setting up the User Store in JBoss on page 158](#)
- [Logging on page 161](#)
- [Enabling Full Text Search on page 168](#)
- [SSL Certificates on page 176](#)
- [Configuring LDAP over SSL/TLS on page 183](#)
- [Configuring for Non-Latin HTTP Parameters on page 184](#)

Launching SOA Systinet on JBoss

The `SOA_HOME/bin` directory of SOA Systinet contains the scripts `serverstart`, `serverstop` and `env-jboss`.

Running `serverstart` calls `env-jboss`, which sets environment parameters for JBoss. Specifically, `env-jboss` gives JBoss access to the SOA Systinet client truststore and optimizes JBoss memory allocation.

In some production environments, where the SOA Systinet is widely distributed or clustered, or when there are applications other than SOA Systinet on the same JBoss server, or where JBoss is using non-default configuration or rmi ports, it might be preferable to use the native JBoss `run` scripts.

In this case, you must first modify the `run` script of each host JBoss as described in [Modifying the JBoss Run Script on page 62](#). The contents of `serverstart` and `serverstop` are also useful guides in this case.

Enabling SSO in JBoss

SOA Systinet automatically configures SSO when SOA Systinet is deployed to a single JBoss application server. For JBoss clusters, the application server requires you to authenticate again every time you request a URL pointing to a previously unaccessed WAR module. (For example, log in to the UI and access a REST endpoint, JBoss requests authentication again).

To prevent this behavior and enable a single login for applications deployed to JBoss clusters, you must change the configuration:

To enable SSO in JBoss clusters:

- 1 Open `JBOSS_HOME/server/CONFIG_NAME/deploy/jboss-web.deployer/server.xml` with a text editor.
- 2 Uncomment the following section:

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn"/>
```
- 3 Save `server.xml` and restart the application server.

For more details about SSO in JBoss, see <http://www.jboss.org/community/docs/DOC-12280>.

Setting up the User Store in JBoss

By default, SOA Systinet uses a JBoss user store to authenticate users. The default user store is a plain text file `JBOSS_PROFILE/conf/users.properties`, which contains lines with `USERNAME=PASSWORD`. All users listed in this file can authenticate with SOA Systinet.

SOA Systinet defines a new JBoss security domain that you can customize to set up authentication against various user stores, including LDAP. The definition of this domain is contained in `SOA_HOME/depot/jboss/hp-soa-systinet.sar`, which is deployed to JBoss during installation.

To modify the JBoss authentication:

- 1 Extract `SOA_HOME/deploy/jboss/hp-soa-systinet.sar` to a directory.
- 2 In the unzipped directory, open `hp-systinet-login-config.xml` with a text editor.
- 3 Change the login module definitions as required.

For details, see the JBoss security documentation.

[Example 5 on page 160](#) is an excerpt of the relevant section of this file.

- 4 Zip the directory back to `hp-soa-systinet.sar`.
- 5 Redeploy the SAR file to `JBOSS_PROFILE/deploy/hp-soa-systinet.sar` and restart JBoss.

Example 5. SOA Systinet JBoss Login Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE policy PUBLIC "-//JBoss//DTD JBOSS Security Config 3.0//EN"
"http://www.jboss.org/j2ee/dtd/security_config.dtd">
<policy>
  <application-policy name="hp-systinet">
    <authentication>
      <!-- ===== -->
      <!-- CLIENT CERT authentication EXAMPLE -->
      <!-- ===== -->
      <!-- JBOSS's SSL client certificate mapping, uncomment when using
            CLIENT-CERT login method -->
      <!--
        <login-module code="org.jboss.security.auth.spi.BaseCertLoginModule" flag="optional">
          <module-option name="password-stacking">useFirstPass</module-option>
          <module-option name="verifier">org.jboss.security.auth.certs.AnyCertVerifier</module-option>
          <module-option name="securityDomain">java:/jaas/hp-systinet</module-option>
        </login-module>
      -->
      <!-- ===== -->
      <!-- USERNAME/PASSWORD authentication EXAMPLE -->
      <!-- ===== -->
      <!-- JBOSS's login module that verifies name and password against users.properties -->
      <!-- file from the classpath (classpath contains JBOSS's configuration conf directory -->
      <login-module code="org.jboss.security.auth.spi.UsersLoginModule" flag="optional">
        <module-option name="password-stacking">useFirstPass</module-option>
      </login-module>
      <!--
        JBOSS's login module that verifies name and password against LDAP.
        To enable LDAP authentication, uncomment the following login-module element,
        customize the module options according to your environment.
        Finally, remove or comment the previous login module org.jboss.security.auth.spi.UsersLoginModule.
      -->
      <login-module code="com.hp.systinet.security.jboss.LdapLoginModule" flag="optional">
        <module-option name="java.naming.provider.url">ldap://localhost:63284</module-option>
        <module-option name="java.naming.factory.initial">com.sun.jndi.ldap.LdapCtxFactory</module-option>

        <module-option name="java.naming.security.authentication">simple</module-option>
        <module-option name="bindDN">uid=user,ou=people,dc=your,dc=company</module-option>
        <module-option name="bindCredential">changeit</module-option>
        <module-option name="baseCtxDN">dc=your,dc=company</module-option>
        <module-option name="baseFilter">(&uid={0})(objectClass=person)</module-option>
        <module-option name="searchScope">SUBTREE_SCOPE</module-option>
      </login-module>
    </authentication>
  </application-policy>
</policy>
```



```

        <module-option name="allowEmptyPasswords">false</module-option>
        <module-option name="password-stacking">useFirstPass</module-option>
    </login-module>
    <!-- ===== -->
    <!-- Mandatory Role Mapping, authenticated users -->
    <!-- will become members of "authenticated" role -->
    <!-- ===== -->
    <!-- custom login module is used to assign authenticated role -->
    <login-module code="com.hp.systinet.security.jboss.AssignRoleLoginModule" flag="optional">
        <module-option name="role">authenticated</module-option>
    </login-module>
</authentication>
</application-policy>
</policy>

```

Logging

SOA Systinet uses log4j logging, a popular logging package for Java that offers various options to change the logging format and/or output.

This section describes the default use of log4j logging by SOA Systinet in the following sections:

- [Log4j Configuration on page 161](#)
- [Default Log4j Configuration on page 162](#)
- [Audit Logging on page 166](#)
- [Modifying JBoss Logging on page 166](#)



You can change the logging options to suit your needs.

For more details, see the log4j manual at <http://logging.apache.org/log4j/1.2/manual.html>.

Log4j Configuration

SOA Systinet relies on the log4j configuration chosen using the "Default Initialization Procedure", described in <http://logging.apache.org/log4j/1.2/manual.html>.

This default initialization procedure results in the following configuration:

- The default logging configuration, as detailed in [Example 6 on page 164](#), is used for the SOA Systinet EAR file deployed to OAS, WebLogic, WebSphere. The file `log4j.properties`, which is contained in the EAR file, contains the default configuration.
- The option, `-Dlog4j.configuration=file:/ABSOLUTE_LOG4J_CONFIG_FILE_PATH`, can be added to the command that starts your application server. This enables you to override the default configuration contained in the EAR file.

SOA Systinet tools execute a java command with a `-Dlog4j.configuration` option that points to a `SOA_HOME/conf/log4j.config`.

SOA Systinet creates log files for these tool executions in the `SOA_HOME/log` directory.

- The logging configuration for an EAR deployed to JBoss is updated during installation, the content of this configuration is similar to the default properties, but is expressed in an XML file.
 - `JBOSS_HOME/server/JBOSS_PROFILE/conf/jboss-log4j.xml`

The audit log file is created in the `JBOSS_HOME/server/JBOSS_PROFILE/log` directory. The Application log is a part of the default JBoss log output (the console and also the `JBOSS_HOME/server/JBOSS_PROFILE/log/server.log` file).

If you are not sure about the logging configuration, do one of the following:

- Use the SOA Systinet Self-Tester, which reports the location of the log4j configuration in use.

For details, see [SOA Systinet Self-Test on page 37](#).

- Add the option `-Dlog4j.debug` to the application server start command and restart the application server.

Log4j then outputs configuration messages to the console.

Default Log4j Configuration

The default log4j configuration from a deployed SOA Systinet is shown in [Example 6 on page 164](#).



SOA Systinet tools use the configuration from `SOA_HOME/conf/log4j.config`, which may be different.

Example 6. Log4j Configuration File

```
# put all logs to console and a log file
log4j.rootLogger=INFO,stdout,file

# console appender
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%p: %c{2} - %m%n

# file appender
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.maxFileSize=20MB
log4j.appender.file.maxBackupIndex=5
log4j.appender.file.File=log4j.log
log4j.appender.file.threshold=INFO
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c - %m%n

# audit log appender
log4j.appender.Systinet_AUDIT=org.apache.log4j.RollingFileAppender
log4j.appender.Systinet_AUDIT.File=hpsoa_audit.log
log4j.appender.Systinet_AUDIT.MaxFileSize=10000KB
log4j.appender.Systinet_AUDIT.MaxBackupIndex=10
log4j.appender.Systinet_AUDIT.layout=org.apache.log4j.PatternLayout
# see http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html
# for formating rules, following extra arguments can be moreover used to
# customize the format
# %X{audit.eventId} - event ID
# %X{audit.result} - event result
# %X{audit.category} - event category
# %X{audit.ctxId} - event context id
# %X{audit.actor} - event actor
# %X{audit.resource} - event actor
# %X{audit.detail} - event detail
log4j.appender.Systinet_AUDIT.layout.ConversionPattern="%d",%X{audit.category}:%X{audit.eventId},
    %X{audit.result},%X{audit.ctxId},"%X{audit.actor}","%X{audit.resource}","%X{audit.detail}%n

# configure audit logging
log4j.category.com.hp.systinet.audit.event=DEBUG,Systinet_AUDIT
log4j.additivity.com.hp.systinet.audit.event=true

# limit categories that are too verbose
log4j.category.org.apache.xml.security=ERROR,file,stdout
```

```
log4j.additivity.org.apache.xml.security=true
log4j.category.org.hibernate=ERROR,stdout,file
log4j.additivity.org.hibernate=true
```

This configuration instructs log4j to do the following:

- 1 Print information, warning, and error messages to the console, and to a file named `log4j.log`, for all logging categories that are not explicitly declared.

SOA Systinet also uses the logging categories which start with one of the following:

- `com.hp.systinet`
- `org.hp.systinet`
- `com.systinet`
- `org.systinet`

- 2 Print the audit log to a file named `hpsoa_audit.log`

The format of the log is specified in the `log4j.appender.Systinet_AUDIT.layout.ConversionPattern` property in [Example 6 on page 164](#). Each audit event is a single line that starts with date and time (formatted according to ISO8601), followed by comma separated attributes of the event.

- 3 A deployed SOA Systinet creates the log files in the following locations:

- **For JBoss:**

```
JBOSS_HOME/server/PROFILE_NAME/log
```

- **For OAS:**

```
OAS_HOME/j2ee/OC4J_INSTANCE_NAME
```

- **For WebLogic:**

```
DOMAIN_HOME
```



DOMAIN_HOME is the location set in [Creating Domains for SOA Systinet on page 84](#).

- **For WebSphere:**

PROFILE_HOME



PROFILE_HOME is the location set in [Creating a Profile on page 100](#).

- 4 The logging category, *com.hp.systinet.audit.event*, is used to log audit events. This logging category also has subcategories according to the audit event category. For example, the logging category name for audit events in the *licensing* category is *com.hp.systinet.audit.event.licensing*.

You can change the output or strip down the audit log for any particular audit category.

- 5 The other declared logging categories (hibernate, apache xml security) are stripped to only log error messages. These categories are too verbose for printing if information messages are also logged (the default for all categories).

Audit Logging

SOA Systinet also uses an audit log to contain events triggered by SOA Systinet functionality. SOA Systinet creates an `hpsoa_audit.log` in the default application server logging directory.

For more details about the content of the audit log and possible configuration, see "Using the Audit Log" in the *HP SOA Systinet Administrator Guide*.

Modifying JBoss Logging

By default, SOA Systinet logs messages to the hosting application server log files. When SOA Systinet is deployed to JBoss, log messages are sent to the following file:

```
JBOSS_DEPLOY\log\server.log
```



`JBOSS_DEPLOY` is the deployment directory on the JBoss application server where SOA Systinet is deployed.

The message threshold level, by default, is `INFO`.

To modify the log file parameters:

- 1 Stop the SOA Systinet server.
- 2 Save `JBOSS_DEPLOY\conf\jboss-log4j.xml.log` to a recoverable backup location in case you need it later.
- 3 Open `JBOSS_DEPLOY\conf\jboss-log4j.xml.log` with a text editor.
- 4 The logging parameter can be one of the following values:
 - `DEBUG`
 - `INFO`
 - `WARNING`
 - `ERROR`

Each value includes the messages for more serious values. For example, setting the logging level to `WARNING` would write all warning and error messages to the log.

To set a logging level, add the following parameter after the `MaxFileSize` parameter:

```
<param name="Threshold" value="LEVEL"/>
```

- 5 You can also set size-based rolling instead of time/date-based rolling.

Comment out or delete the time/date based rolling appender:

```
<!-- A time/date based rolling appender -->
<appender name="FILE" class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="{jboss.server.log.dir}/server.log"/>
  <param name="Append" value="false"/>
  <!-- Rollover at midnight each day -->
```

```

<param name="DatePattern" value="'.'yyyy-MM-dd"/>
<!-- Rollover at the top of each hour
<param name="DatePattern" value="'.'yyyy-MM-dd-HH"/>
-->
<layout class="org.apache.log4j.PatternLayout">
  <!-- The default pattern: Date Priority [Category] Message\n -->
  <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
  <!-- The full pattern: Date MS Priority [Category] (Thread:NDC) Message\n
  <param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x) %m%n"/>
  -->
</layout>
</appender>

```

6 To enable size-based rolling, uncomment the size-based rolling appender:

```

<!-- A size based file rolling appender
<appender name="FILE" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.log.dir}/server.log"/>
  <param name="Append" value="false"/>
  <param name="MaxFileSize" value="500KB"/>
  <param name="MaxBackupIndex" value="1"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
  </layout>
</appender>
-->

```

7 In the size-based appender section, set the maximum file size to for example, 100KB, by modifying the *MaxFileSize* parameter:

```

<param name="MaxFileSize" value="100KB"/>

```

8 Save `JBOSS_DEPLOY\conf\jboss-log4j.xml.log`.

9 Start the SOA Systinet server.

Enabling Full Text Search

The SOA Systinet full text search is an optional feature based on relational database extensions.

To enable FTS:

- 1 Prepare FTS on the database server:
 - a Create an index for column "m_extension" from "ry_resource" table.
 - b Create an index for column "data" from "ry_resource" table.
 - c Schedule update of these indexes.

- 2 Activate FTS in the SOA Systinet UI, as described in "SOA Systinet Configuration Options" in the *HP SOA Systinet Administrator Guide*.

The following sections provide details on enabling Full Text Search:

- [Enabling Full Text Search on DB2 on page 169](#)
- [Enabling Full Text Search on MSSQL on page 171](#)
- [Enabling Full Text Search on Oracle on page 173](#)

By default, SOA Systinet appends a % to search terms. To disable this functionality after installation, see "Disabling the Addition of % to Search Terms" in the *HP SOA Systinet Administrator Guide*.

Make sure that your database server meets the system requirements described in the [Preparing Databases on page 40](#).

Enabling Full Text Search on DB2

To enable full text search you must create indexes and schedule their update in DB2. Use the DB2 Net Search Extender. Connect to the database using the same credentials used during installation.

Follow [Example 7 on page 170](#).

Example 7. Create Indexes for FTS and Schedule Synchronization in DB2

```
db2text START

#use sa user in this case
db2text ENABLE DATABASE FOR TEXT CONNECT TO <database> USER sa USING <password>

db2text CREATE INDEX idx_ry_resource_meta FOR TEXT ON ry_resource(m_extensions)
CONNECT TO <database> USER <user> USING <password>

db2text CREATE INDEX idx_ry_resource_data FOR TEXT ON ry_resource(data)
CONNECT TO <database> USER <user> USING <password>

#schedule a regular index update each day at midnight
db2text ALTER INDEX idx_ry_resource_meta FOR TEXT UPDATE FREQUENCY D(*) H(0) M(0)
CONNECT TO <database> USER <user> USING <password>

db2text ALTER INDEX idx_ry_resource_data FOR TEXT UPDATE FREQUENCY D(*) H(0) M(0)
CONNECT TO <database> USER <user> USING <password>
```

Commands to update the index manually can be found in [Example 8 on page 170](#).

Example 8. Synchronizing Indexes in DB2 Manually

```
db2text UPDATE INDEX idx_ry_resource_meta FOR TEXT
CONNECT TO <database> USER <user> USING <password>

db2text UPDATE INDEX idx_ry_resource_data FOR TEXT
CONNECT TO <database> USER <user> USING <password>
```

For more scheduling details, see the *DB2 Net Search Extender* documentation.

Searching Uploaded Documents with DB2

DB2 Net Search Extender uses external *Stellent/Oracle Outside In* technology to obtain text from DOC, PDF, and other text format files before indexing. For details, see the *Net Search Extender Administration and User Guides*.



Indexing DOC and PDF files in SOA Systinet with DB2 has not been tested.

Enabling Full Text Search on MSSQL

To enable full text search you must enable the service and create a full text catalog and indexes. Use MSSQL Server Management Studio or the sqlcmd command line tool.

Connect to the database using the same parameters used during SOA Systinet installation.

To enable full text search on MSSQL:

- 1 Make sure that the SQL Server Fulltext Search service is running, and that the database is full-text enabled.

By default, new databases are full-text enabled unless you create them with MSSQL Server Management Studio.

In this case, select the database in the Object Explorer window, and select **Properties**→**Files**, and then select **Use full-text indexing**.

- 2 To create a full-text catalog, execute the following command:

```
sqlcmd -U <user> -P <password> -d <database>  
CREATE FULLTEXT CATALOG ry_resource_ftsc  
go
```



You must have CREATE FULLTEXT CATALOG permission.

It is possible to reuse an existing catalog, but HP Software recommend creating a new one for independent management purposes.

For more details, see <http://msdn2.microsoft.com/en-us/library/ms189520.aspx>.

- 3 Do one of the following:

- To create a full-text index that is synchronized immediately after any data changes, execute the following command:

```
sqlcmd -U <user> -P <password> -d <database>
CREATE FULLTEXT INDEX ON ry_resource(
    m_extensions TYPE COLUMN m_extensions_fe LANGUAGE 0x0,
    data TYPE COLUMN data_fe LANGUAGE 0x0)
KEY INDEX pk_resource_12 ON ry_resource_ftsc WITH CHANGE_TRACKING AUTO
go
```

- To create a full-text index that is synchronized manually, execute the following command:

```
sqlcmd -U <user> -P <password> -d <database>
CREATE FULLTEXT INDEX ON ry_resource(
    m_extensions TYPE COLUMN m_extensions_fe LANGUAGE 0x0,
    data TYPE COLUMN data_fe LANGUAGE 0x0)
KEY INDEX pk_resource_11 ON ry_resource_ftsc WITH CHANGE_TRACKING OFF, NO POPULATION
go
```

For more details, see <http://msdn2.microsoft.com/en-us/library/ms187317.aspx>.

To synchronize the index manually, execute the following command:

```
sqlcmd -U <user> -P <password> -d <database>
ALTER FULLTEXT INDEX ON ry_resource START FULL POPULATION
go
```

The statement executes asynchronously, so the population may take some time.

To verify the population status, execute the command:

```
SELECT FULLTEXTCATALOGPROPERTY('ry_resource_ftsc', 'PopulateStatus')
go
```

Index population is complete when the population status is 0.

For more details, see <http://msdn.microsoft.com/en-us/library/ms188359.aspx>.

Searching Uploaded Documents with MSSQL

MSSQL uses iFilters to create text streams from various text formats. The iFilter for DOC files is already included, but you must install the iFilter for PDF if you want to search PDF files.

To install a PDF iFilter:

1 Install one of the following PDF iFilters:

- Foxit provides an iFilter for 32/64-bit systems (recommended).

For details, go to <http://www.foxitsoftware.com/pdf/ifilter>.

- Adobe Reader 9 includes an iFilter for 32-bit systems.

For details, go to <http://adobe.com>.

2 To load an external iFilter, execute the following statements in MSSQL using the administrator account:

```
exec sp_fulltext_service 'load_os_resources', 1;
```

```
exec sp_fulltext_service 'verify_signature', 0;
```

```
go
```

3 Restart MS SQL Server including the MS SQL Server FullText Search.

4 To verify the iFilter, execute the following command using the administrator account:

```
select * from sys.fulltext_document_types where document_type='.pdf'
```

5 Recreate fulltext indexes. You can view the available languages with the following command:

```
select * from sys.fulltext_languages
```

Enabling Full Text Search on Oracle

To enable full text search, you must create indexes and schedule their update. Use the Oracle **sqlplus** console. Connect to the database using the same credentials used during installation.

The procedure in commands is shown in [Example 9 on page 174](#). It also shows how to synchronize indexes every midnight.



The database user does not have permission to create FTS indexes by default and must be given that permission.

Example 9. Preparing Oracle For Full Text Search using the Scheduling Mechanism

```
sqlplus system/password@connect_identifier
-- add permission to create indexes
GRANT EXECUTE ON "CTXSYS"."CTX_DDL" TO user;
-- add "create job" permission to <user>
GRANT CREATE JOB TO user;
exit;

sqlplus user/password@connect_identifier
CREATE INDEX idx_ry_resource_meta ON ry_resource(m_extensions)
  INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
  ('FILTER CTXSYS.NULL_FILTER SECTION
   GROUP CTXSYS.NULL_SECTION_GROUP
   SYNC (EVERY "TRUNC(SYSDATE)+1") TRANSACTIONAL');

CREATE INDEX idx_ry_resource_data ON ry_resource(data)
  INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
  ('FILTER CTXSYS.NULL_FILTER SECTION
   GROUP CTXSYS.NULL_SECTION_GROUP
   SYNC (EVERY "TRUNC(SYSDATE)+1") TRANSACTIONAL');
```

To enable full text search of pdf, doc, and other document types, use `AUTO_FILTER` in the definition of the `idx_ry_resource_data` index"

```
CREATE INDEX idx_ry_resource_data ON ry_resource(data)
  INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
  ('FILTER CTXSYS.AUTO_FILTER');
```



Do not implement index synchronization `ON COMMIT`. It can cause Oracle thread termination, returning the error message `ORA-error stack (07445[ACCESS_VIOLATION])` logged in `filename.log`. (Tested on Oracle 10gR2 - 10.2.0.1). Use regular synchronization together with the `TRANSACTIONAL` parameter. For details, see [Example 9 on page 174](#).

For more information about creating indexes, see the Oracle documentation at http://download-uk.oracle.com/docs/cd/B19306_01/text.102/b14218/toc.htm



Not all document types can be indexed correctly.

For details, see http://download.oracle.com/docs/cd/B19306_01/text.102/b14218/afilsupt.-htm#i634493.

Synchronizing Indexes

Executing index synchronization manually is shown in [Example 10 on page 175](#).

Example 10. Synchronizing Indexes in Oracle Manually

```
sqlplus user/password@connect_identifier
CALL CTX_DDL.SYNC_INDEX('idx_ry_resource_meta', '2M');
CALL CTX_DDL.SYNC_INDEX('idx_ry_resource_data', '2M');
```

Creating an Indexing Stoplist

You can optionally manage a stoplist by removing words that could frequently appear in documents. By default, the Oracle index stoplist includes words such as "to". Full-text searches including these words return a false empty result. Alternatively, the database administrator should provide SOA Systinet users with the stoplist, and a warning not to use these terms in full-text searches.

An example of commands to set up a stoplist on Oracle is shown in [Example 11 on page 176](#).

Example 11. Creating an Oracle Indexing Stoplist

```
call CTX_DDL.CREATE_STOPLIST('MyStoplist');
call CTX_DDL.ADD_STOPWORD('MyStoplist', 'a');
... Add a word that should not be indexed. Repeat the command for each word to be excluded.

-- Include the DROP INDEX commands only if an index already exists.
DROP INDEX idx_ry_resource_meta;
DROP INDEX idx_ry_resource_data;
CREATE INDEX idx_ry_resource_meta ON ry_resource(m_extensions) INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
('filter CTXSYS.NULL_FILTER SECTION GROUP CTXSYS.NULL_SECTION_GROUP STOPLIST MyStoplist
SYNC (EVERY "TRUNC(SYSDATE)+1") TRANSACTIONAL') ;
CREATE INDEX idx_ry_resource_data ON ry_resource(data) INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
('filter CTXSYS.NULL_FILTER SECTION GROUP CTXSYS.NULL_SECTION_GROUP STOPLIST MyStoplist
SYNC (EVERY "TRUNC(SYSDATE)+1") TRANSACTIONAL');
```

SSL Certificates

SOA Systinet can use HTTPS for communicating with other applications, such as UDDI registries, *HP Business Availability Center*, and *HP SOA Manager*, or to publish resources, such as WSDL and XML schemas, accessible via HTTPS.

HTTPS runs on top of SSL and requires additional set up steps. SSL requirements are driven by the target server, which can operate in one of the following modes:

1 One-way SSL

Mostly used for WWW, where the server authenticates to the client by providing its certificate.

The following conditions must be met:

- The server certificate must be valid and trusted.
- The resource HTTPS URL must contain a host name contained in the target server SSL certificate.

2 Required two-way SSL

Mutual authentication is required and both server and client authenticate each other using certificates.

The following conditions must be met:

- The requirements for one-way SSL.
- The client must provide its certificate, and the server must trust the client certificate.

3 Wanted two-way SSL

Mutual authentication is wanted but not mandatory and the connection can be established as one-way SSL.

The requirements are the same as for one-way SSL.

Using SSL is described in the following sections:

- [Identifying Server SSL Requirements on page 177](#)
- [SSL Server Certificate Trust on page 179](#)
- [Trusting HP SOA Registry Foundation Certificates on page 179](#)
- [Importing Client Certificates for Two-Way SSL on page 179](#)
- [SSL Customization on page 179](#)
- [SSL Troubleshooting on page 182](#)

Identifying Server SSL Requirements

SOA Systinet deployed to an application server acts as an HTTPS client.

The SSL requirements of the target server can be determined using the `ssltool` provided with SOA Systinet.

For example, with the command:

```
ssltool serverInfo --url https://localhost:8443 --certFile localhost.crt
```

The output resembles the following:

```
localhost:2968  
Host: localhost/127.0.0.1  
Port: 2968  
Client authentication: WANTED
```

```

Accepted issuers of client certificates:
  OU=Some CA, CN=CA
Server Certificate:
  Subject: CN=localhost
  Issuer: OU=Some CA, CN=CA
  Issued for hostname(s): localhost
  Algorithm: RSA
  Valid from: 12/7/07 4:22 PM
  Valid to: 12/4/17 4:22 PM
Received Server Certificate Chain:
  CN=localhost
  OU=Some CA, CN=CA
PEM Encoded Received Server Certificate Chain:
-----BEGIN CERTIFICATE-----
  ...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
  ...
-----END CERTIFICATE-----

```

The output contains the following elements:

- The SSL server mode is output after `Client Authentication`, where:
 - `NOT_REQUESTED` means one-way SSL.
 - `WANTED` means non-mandatory two-way SSL.
 - `REQUIRED` means required two-way SSL.
- The accepted issuers for two-way SSL. These display the subject names of accepted certificate authorities that can issue client certificates.
- The server certificate details saved to a file, `localhost.crt`.
- Host names that must be used in URLs are listed after `Issued for hostname(s):`.
- PEM encoded certificates that must be trusted.

For more details about the SSL Tool, see "SSL Tool" in the *HP SOA Systinet Administrator Guide*.

SSL Server Certificate Trust

By default, any SSL server certificate is trusted.



For non-default options, see [SSL Customization on page 179](#).

Trusting HP SOA Registry Foundation Certificates

If you are using SOA Systinet integrated with HP SOA Registry Foundation and intend to import or export artifacts and taxonomies, then SOA Systinet must trust the HP SOA Registry Foundation certificate.

For details, see [SSL Server Certificate Trust on page 179](#).

Importing Client Certificates for Two-Way SSL

A client certificate (a private key with an associated certificate) issued by a certificate authority trusted by the server must be imported to the keystore. By default, this keystore is the one used by JDK. Usually it is empty or unspecified. The location of the JDK SSL keystore is determined using a procedure explained at <http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html#X509KeyManager>. JDK keystores are managed by the keytool.

For details about keytool, see <http://java.sun.com/j2se/1.5.0/docs/tooldocs/index.html#security>.



For non-default options, see [SSL Customization on page 179](#).

SSL Customization

SOA Systinet customizes SSL trust, key management, and hostname verification. There are four available SSL customizations available after installation. The default customization is named `skipped` and this is the customization used in the previous procedures in this section.

The list of available customizations is available with the command:

```
ssltool customize --info
```

The output should resemble the following:

```
Effective customization: composite
Available customizations:
--> default
    Java/JSSE default key/trust stores, default hostname verifier.
--> skipped
    Server certificates are always trusted and pass hostname verification, default
    key manager is used.
--> database
    Database key/trust stores, default hostname verifier.
--> composite
    Composition of database and default key/trust stores, default hostname verifier.
```

- **default**

This means no customization and the application server trust and keystore is used.

Hostname verification is required.

- **skipped**

This means that server certificates are ignored as no trust is required.

Hostname verification is ignored.

Client certificates are taken from a file described here at <http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html#X509KeyManager>.

The `skipped` customization is a default, it is chosen during installation by doing one of the following:

- In the GUI Installation wizard, Endpoint Properties page, deselect **Verify Certificates**.

For details, see [Using the GUI Installer on page 19 Step 18](#).

- In the `configuration-properties.xml` file, set the following property:

```
platform.certVerification=skipped
```

For details, see [Configuring the Endpoint on page 133](#).

- **database**

This means that only the SOA Systinet database is used for the trust and keystore.

Hostname verification is required.

- **composite**

This means that a composite of the JDK truststore and SOA Systinet database truststore is used.

A certificate is trusted if it is in either the JDK or the SOA Systinet database truststore.

Client certificates are taken from both the JDK and SOA Systinet database keystores.

Hostname verification is required.

The `composite` customization can be chosen during installation by doing one of the following:

- In the GUI Installation wizard, Endpoint Properties page, select **Verify Certificates**.

For details, see [Using the GUI Installer on page 19 Step 18](#).

- In the `configuration-properties.xml` file, set the following property:

```
platform.certVerification=composite
```

For details, see [Configuring the Endpoint on page 133](#).

You can manage the database keystore and truststore using the `ssltool` provided with SOA Systinet. These stores are specific to SOA Systinet. Unlike the JDK keystore, changing the database keystore does not require a restart of the application server, because changes take effect in approximately thirty seconds.

To import a server certificate to the database truststore:

- Execute the following command:

```
SOA_HOME/bin/ssltool keystoreEI -i truststore --certfile CERTIFICATE_FILE --add
```

To import a client certificate to the database keystore:

- Execute the following command:

SOA_HOME/bin/ssltool keystoreEI -i keystore --keystore CLIENT_CERT --storepass PASSWORD --add

CLIENT_CERT can be either a Java keystore with a private key or a p12 file (PKCS 12). *PASSWORD* is the password used to protect the file.

For more details about the SSL Tool, see "SSL Tool" in the *HP SOA Systinet Administrator Guide*.

To change the SSL customization:

- Do one of the following:
 - Change the setting in the SOA Systinet Configuration page.

For details, see "Configuration Options" in the *HP SOA Systinet Administrator Guide*.

- Execute the following command:

SOA_HOME/bin/ssltool customize --change CUSTOMIZATION_NAME

SSL Troubleshooting

You can test the SSL set up for a particular HTTPS URL using the self-tester.

The test page is available at the following URL:

http://localhost:port/context/self-test/self-http-test

Type the URL to check and click **Test Connection**.

The output resembles the following:

URL:

Tested url: <https://localhost>

Result:

```
javax.net.ssl.SSLPeerUnverifiedException: peer not authenticated
* * *
```

```
Server certificate details
Subject DN: CN=CA
Issuer (CA) DN: CN=CA
Valid from: Fri Mar 14 16:04:32 CET 2008
Valid to: Mon Mar 12 16:04:32 CET 2018
Serial Number: 86
Server certificate chain
* * *
```

You can download server certificate [here](#)

For more details about the self-tester, see [SOA Systinet Self-Test on page 37](#).

Configuring LDAP over SSL/TLS

You can configure LDAP over *SSL* (or *TLS*) with a directory server of your choice. HP Software recommend that you first install SOA Systinet with a connection to LDAP that does not use SSL. You can then verify the configuration by logging in as a user defined in this directory before configuring use of SSL.

The configuration procedure assumes that you have already installed SOA Systinet with an LDAP account provider.

SOA Systinet must not be running.

LDAP over SSL Without Client Authentication

In this case only LDAP server authentication is required. This is usually the case.

To change the LDAP configuration, run the Setup Tool and change **Naming Provider URL** to use the `ldaps` protocol and the port on which the directory server accepts SSL/TLS connections. An example of such a URL is, `ldaps://ldap.test.com:636`.

Make sure that the hostname specified in the `java.naming.provider.url` property matches the name that is in the directory server certificate's subject common name (CN part of certificate's Subject). Otherwise you get an exception during startup of SOA Systinet. It informs you of a hostname verification error. The stacktrace contains the hostname that you must use.

LDAP over SSL With Mutual Authentication

SOA Systinet does not support LDAP over SSL with mutual authentication.

Ensuring Trust with the LDAP Server

The client that connects to the SSL/TLS server must trust the server certificate in order to establish communication with that server. The configuration of LDAP described in this section inherits the default rule for establishing trust from JSSE (the Java implementation of SSL/TLS). This is based on trust stores.

For details, see [SSL Server Certificate Trust](#) on page 179.

Configuring for Non-Latin HTTP Parameters

The application server must be configured to handle non-Latin characters in HTTP parameters on behalf of SOA Systinet.

The application specific procedures are described in the following sections:

- [Configuring HTTP Encoding for JBoss](#) on page 184
- [Configuring HTTP Encoding for WebSphere](#) on page 185
- WebLogic and OAS configuration is handled using descriptor files in the SOA Systinet EAR.

Configuring HTTP Encoding for JBoss

If you deploy the SOA Systinet EAR to JBoss manually then make the following changes to allow non-Latin characters in HTTP parameters:



This process is automated when the installer deploys the EAR file to JBoss.

To enable non-Latin encoding for JBoss:

- 1 Open `JBOSS_HOME/server/CONFIGURATION/deploy/jboss-web.deployer/server.xml` with a text editor.

2 In all `connector` elements defined in `server.xml`, set the `URIEncoding` attribute to `UTF-8`.

Configuring HTTP Encoding for WebSphere

When you deploy the SOA Systinet EAR to WebSphere then make the following changes to allow non-Latin characters in HTTP parameters:

To enable non-Latin encoding for WebSphere:

- Configure the application server by adding the following switches to generic JVM arguments:

```
Dclient.encoding.override=UTF-8 -Ddefault.client.encoding=UTF-8
```

For more details, see the WebSphere documentation at the following location:

http://publib.boulder.ibm.com/infocenter/pvcwesd/index.jsp?topic=/com.ibm.websphere.west.-doc/server/srvr_exp_override.html