HP SOA Systinet

Software Version: 2.52

Installation and Deployment

Document Release Date: November 2007
Software Release Date: November 2007

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Third-Party Web Sites

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Copyright Notices

Trademark Notices

Java™ is a US trademark of Sun Microsystems, Inc. Microsoft®, Windows® and Windows XP® are U.S. registered trademarks of Microsoft Corporation. IBM®, AIX® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries. BEA® and WebLogic® are registered trademarks of BEA Systems, Inc.

Contents

Welcome to This Guide. 7
How This Guide Is Organized. 7
Document Conventions. 8
Documentation Updates. 9
Support. 10

I Before Installing. 13

1 Prerequisites. 15
Hardware. 15
Software. 15

2 Supported Platforms. 17

3 Designing Your Deployment. 19
Development. 19
Production. 19
Sizing. 21

II Database Administration. 23

4 Database Setup Overview. 25
Setting Up DB2. 25
Setting Up Oracle. 25

5 Database Installation Types. 27

6 Manual Database Arrangement. 31
Setting Up The "power_user" Account. 31
Creating Schema Manually. 33

Running SOA Systinet With Minimal Privileges. 33
When Extensions Are Applied. 37

III Deploying SOA Systinet To Application Servers. 39

7 Installing SOA Systinet on JBoss. 41
High-level JBoss Installation Procedure. 41
Clustered JBoss. 42
Setting Up the JBoss Server. 48
Launching SOA Systinet on JBoss. 57

8 Deploying SOA Systinet to WebLogic. 59
Creating a Domain. 59
Setting Up Trust on WebLogic. 65
Creating JDBC Resources. 67
Creating JMS Resources. 70
Creating a Mail Session. 77
Deploying SOA Systinet Components. 77
Starting and Verifying SOA Systinet. 78

9 Deploying SOA Systinet to WebSphere. 81
Creating a Profile. 82
Setting Up Trust on WebSphere. 82
Creating a Mail Session. 83
Creating JDBC Resources. 83
Deploying the SSO Service and Creating EAR Files. 87
Setting Up log4j Logging, Memory Allocation and Java Properties. 89
Creating a Messaging Bus. 91
Setting Up JMS for the Reporting Service. 92
Setting Up JMS for Platform. 95
Setting Up JMS for Policy Manager. 97
Deploying the Remaining EAR Files. 99

10 Deploying SOA Systinet to Oracle Application Server. 101

Update JDBC Driver. . . . 102
Setting Up SSL on OAS. . . . 102
Allowing JNDI Lookups. . . . 105
Setting log4j Properties. . . . 106
Creating JDBC Resources. . . . 107
Allocating Memory and Setting Java Options. . . . 110
Creating JMS Resources. . . . 110
Creating and Deploying SOA Systinet EARs. . . . 111

IV Installing SOA Systinet Components. . . . 115

11 Installing the Single Sign-On Service. . . . 117
Running the SSO Installer. . . . 117
LDAP Accounts Integration. . . . 121

12 Installing Reporting Service. . . . 129

13 Installing Platform. . . . 135

14 Installing Policy Manager. . . . 143

15 Using Silent Installation. . . . 149

V After Installation. . . . 151

16 Self-Testers. . . . 153

17 Enabling Full Text Search. . . . 157
Creating Full Text Search Indexes on DB2. . . . 158
Creating Full Text Search Indexes on Oracle. . . . 159
Disabling the Addition of % to Search Terms. . . . 160

18 Importing SOA Systinet Registry Certificate. . . . 163
Importing Systinet Registry Certificate to JBoss. . . . 163
Importing Systinet Registry Certificate to WebLogic. . . . 164
Importing Systinet Registry Certificate to WebSphere. . . . 164

Importing Systinet Registry Certificate to Oracle Application Server. 164
19 Proxy Setup. 165
Index. 169

Welcome to 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.

How This Guide Is Organized

HP SOA Systinet Installation Guide describes the prerequisites and process of installing HP SOA Systinet to your enterprise.

It contains the following parts:

- **Part I, "Before Installing".** Preparing your enterprise system for HP SOA Systinet
- **Part III, "Deploying SOA Systinet To Application Servers".** A guide to setting up the J2EE application servers hosting SOA Systinet and deploying the SOA Systinet EAR files.
- **Part IV, "Installing SOA Systinet Components".** A guide to using the SOA Systinet installers. These installers can automatically deploy SOA Systinet components to the JBoss application server. For other servers, they create EAR files which you deploy using the application server's tools.
- **Part V, "After Installation".** A guide to the likely next steps and where to find the information required

Document Conventions

The typographic conventions used in this document are:

run.bat make	Script name or other executable command plus mandatory arguments.
[--help]	A command-line option.
either | or	A choice of arguments.
replace_value	A command-line argument that should be replaced with an actual value.
{arg1 | arg2}	A choice between two command-line arguments where one or the other is mandatory.
`rmdir /S /Q System32`	Operating system commands and other user input that you can type on the command line and press **Enter** to invoke. Items in *italics* should be replaced by actual values.
`C:\System.ini`	Filenames, directory names, paths and package names.
`a.append(b);`	Program source code.
`server.Version`	An inline Java or C++ class name.
`getVersion()`	An inline Java method name.
Shift-N	A combination of keystrokes.
Service View	A label, word or phrase in a GUI window, often clickable.
OK	A button in a GUI window.
New->Service	Menu choice.

Documentation Updates

This manual's title page contains the following identifying information:

- Software version number
- 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://ovweb.external.hp.com/lpe/doc_serv/

Support

Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (**http://h20230.www2.hp.-com/svi_partner_list.jsp**), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support Web site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the Mercury Customer Support Web site at: **http://hp.-com/go/hpsoftwaresupport**.
- For the latest information about support processes and tools available for products formerly produced by Systinet, we encourage you to visit the Systinet Online Support Web site at: **http://www.systinet.-com/support/index**.
- If you have additional questions, contact your HP Sales Representative.

HP Software Support

You can visit the HP Software Support Web site at:

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

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

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts

- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To find more information about access levels, 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

Part I. Before Installing

Before installing SOA Systinet. check that you meet system requirements, design your deployment and set up your database.

This part contains the following chapters:

- Prerequisites on page 15. Hardware and software requirements.
- Supported Platforms on page 17. The application servers and databases supported by SOA Systinet, organized by operating system. Please note that minimum version numbers, including service packs/updates etc, are required.
- Designing Your Deployment on page 19. Determining cluster sizes, etc depending on the environment in which you are installing SOA Systinet.

1 Prerequisites

The following hardware and software is required for running SOA Systinet:

Hardware

Hardware requirements vary depending on sizing and deployment type (see Designing Your Deployment on page 19). For a distributed, production environment, the requirements are:

- 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 JDK and a J2EE application server from the list in Supported Platforms on page 17. 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.
- Access to a supported database from Supported Platforms on page 17.

2 Supported Platforms

The following tables present combinations of application servers, JDKs and backend databases that we support. One table is given for each supported operating system. These tables are:

- Table 1 on page 17
- Table 2 on page 18
- Table 3 on page 18
- Table 4 on page 18
- Table 5 on page 18

Table 1. Supported Platforms on Windows

Windows Server 2003 and XP SP 2 on x86 Platform		
J2EE Application Server	**Java JDK**	**Relational Database**
JBoss® 4.0.5	Sun JDK 1.4.2_13 or later, 1.5.0_09 or later	Oracle 10g
BEA® WebLogic Server ® 9.2.2	Sun JDK 1.5.0_09 or later	Oracle 10g
IBM WebSphere® 6.1.0.3	IBM JDK 1.5	DB2 version 9.1
Oracle Application Server 10.1.3.3	Sun JDK 1.5.0_09 or later	Oracle 10g

Table 2. Supported Platforms on Linux

RedHat Enterprise Linux 4.0 ES on x86 Platform		
J2EE Application Server	**Java JDK**	**Relational Database**
JBoss® 4.0.5	Sun JDK 1.4.2_13 or later, 1.5.0_09 or later	Oracle 10g
BEA® WebLogic Server ® 9.2.2	Sun JDK 1.5.0_09 or later	Oracle 10g
IBM WebSphere® 6.1.0.3	IBM JDK 1.5	Oracle 10g
Oracle Application Server 10.1.3.3	Sun JDK 1.5.0_09 or later	Oracle 10g

Table 3. Supported Platforms on Solaris

Solaris 10 on Sparc Platform		
J2EE Application Server	**Java JDK**	**Relational Database**
JBoss® 4.0.5	Sun JDK 1.4.2_13 or later, 1.5.0_09 or later	Oracle 10g
BEA® WebLogic Server ® 9.2.2	Sun JDK 1.5.0_09 or later	Oracle 10g

Table 4. Supported Platforms on AIX

AIX 5L 5.3 on PowerPC		
J2EE Application Server	**Java JDK**	**Relational Database**
IBM WebSphere® 6.0.1.2	IBM JDK 1.4	DB2 version 8.1.1.9
IBM WebSphere® 6.1.0.3	IBM JDK 1.5	DB2 version 9.1

Table 5. Supported Platforms on HP-UX

HP-UX 11.23 and 11.31 on Itanium		
J2EE Application Server	**Java JDK**	**Relational Database**
JBoss® 4.0.5	HP JDK 1.5.0_09 or later	Oracle 10g
BEA® WebLogic Server ® 9.2.2	HP JDK 1.5.0_09 or later	Oracle 10g
Oracle Application Server 10.1.3.3	HP JDK 1.5.0_09 or later	Oracle 10g

3 Designing Your Deployment

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

- **Development.** To evaluate the product, you can deploy it on a single machine.
- **Production.** To use SOA Systinet in a production environment, cluster it with its components installed on separate nodes.

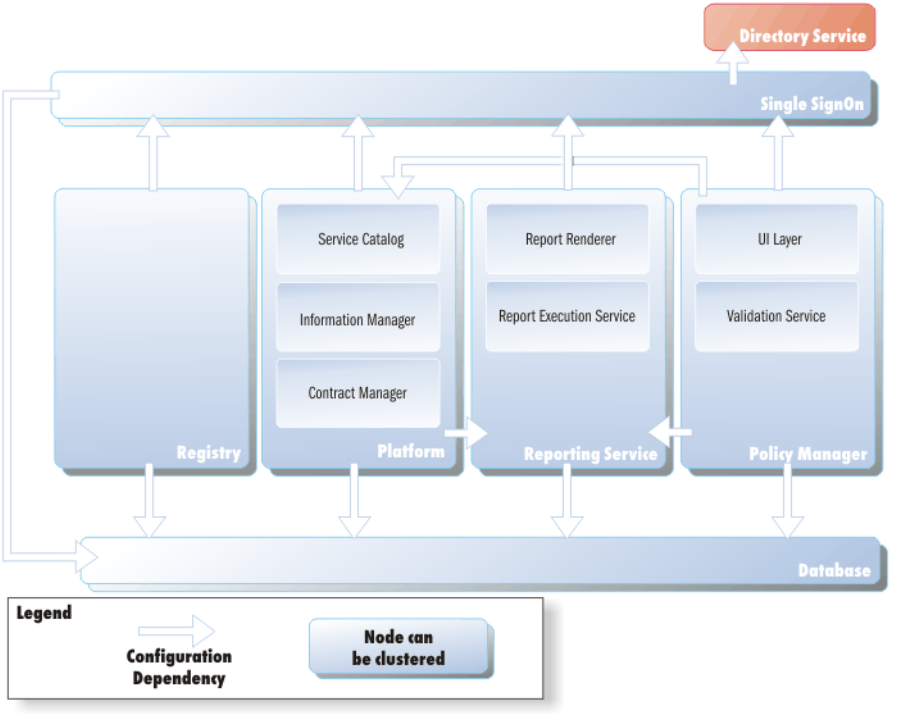
Development

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 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 can only create EAR files, which you then deploy using the application server's tools. You also have to modify server classpaths, configure JMS and set Java properties yourself.

Production

Deploying SOA Systinet for use in a production environment is complex. Different components of SOA Systinet are installed to different machines, and are likely to be clustered as well. A schematic of a production deployment is shown in Figure 1. 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.

Figure 1. Production Environment Deployment



When you deploy SOA Systinet to a production environment you might need to install the SOA Systinet components non-interactively or install them using a reuseable installation configuration. The installation wizard generates an XML property file with the installer configuration. This file can be edited and used in a non-interactive installation.

SSO, Reporting, Platform, and Policy Manager must share the same database.

Sizing

The number of users per physical node depends on whether all SOA Systinet components are installed to the same node or whether SOA Systinet is distributed with each component hosted on its own standalone server or cluster. The numbers are as follows:

Table 6. Users per Physical Node

Components Installed to Same Node	
Type of core	**Users per core**
32-bit Xeon 3.0GHz	20
64-bit Xeon 3.0GHz	35
Distributed Installation	
Type of core	**Users per core**
Xeon 6160 3.0GHz	70

The database requires at least a 2x dualcore Xeon 5160 3.0GHz core and 10,000 RPM disks for every 200 users.

Minimum heap configuration follows. If any server runs out of memory, increase the heap sizes.

- MaxPermSize at least 128MB
- Max heap size per processor:
  - SSO: 512 MB
  - PM: 1368 MB
  - Platform: 768 MB
  - Reporting: 768 MB
- On multiple processor machines using Sun JDK, set `-XX:+UseParallelGC`.

JDBC connection pool size should be at least equal to the number of expected concurrent users. HTTP thread size should be at least equal to 2 times the number of expected concurrent users.

Part II. Database Administration

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

This part contains the following chapters:

- Database Setup Overview on page 25. General instructions on how to prepare the database prior to installing SOA Systinet.
- Database Installation Types on page 27. The database operations supported by SOA Systinet installers and the Setup tool.
- Manual Database Arrangement on page 31. How to manually perform the database operations supported by SOA Systinet installers and the Setup tool, to comply with your security and deployment policies.

4 Database Setup Overview

The databases hosting SOA Systinet repository data must be set up correctly before you install SOA Systinet components. The following sections describe how to set up the database:

- Setting Up DB2 on page 25
- Setting Up Oracle on page 25

Setting Up DB2

Configure the DB2 database as follows for use with SOA Systinet:

- If you plan to use the SOA Systinet full text search feature, make certain the optional DB2 Net Search Extender is installed.
- If one does not exist, create a database that uses the UTF-8 Code Set.
- If it does not exist, create a bufferpool with a minimum page size of 16k. Create a *system temporary tablespace* and a *user temporary tablespace*, each with at least 16k page size, using the bufferpool.
- Create an OS account to hold SOA Systinet tables the database. This must be a a "power_user" account that can create database resources. If SOA Systinet should not connect to the database using the power_user account, create a "common_user" account with minimal privileges, including permission to create alias/synonym. (See Manual Database Arrangement on page 31.)

Note: After installation, If SOA Systinet Full Text Search will be enabled, set up indexes on DB2 as described in Creating Full Text Search Indexes on DB2 on page 158.

Setting Up Oracle

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

- If you plan to use SOA Systinet's Full Text Search feature, include the "Oracle Text" extension when installing the Oracle server. The "Oracle Text" extension is applied to Oracle by default.
- Create a database that uses the Unicode (AL32UTF8) character set and local character set. UTF-8 is the preferred encoding.
- If you set the default JVM language to a language other than English, you *must* use JDBC driver version 10.1.0.2. Otherwise it is not possible to deploy any SOA Systinet components.

After installation, if SOA Systinet Full Text Search will be enabled, set up Oracle indexing as described in Creating Full Text Search Indexes on Oracle on page 159.

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>;
```

5 Database Installation Types

There are three ways to arrange the database and schema for SOA Systinet:

- **Create Database** option in the installation wizards. This option automizes database arrangement as much as possible. On the other hand, it requires database administrator credentials.
- **Create Schema** option in the installation wizards. This option requires the credentials of a "power_user" who can create schema items (tables, indexes and sequences). The database administrator must create this power_user manually. SOA Systinet is installed using this power_user account and the SOA Systinet installation wizards create the schema items.
- Manual database arrangement. If the SOA Systinet administrator does not have power_user credentials, the database administrator must create all schema items and grant access to these items to a common_user account. Manual database arrangement is not the subject of this chapter. See Manual Database Arrangement on page 31.

This chapter describes how to create a new database or a schema in an existing database using either the SOA Systinet installer or SOA Systinet Setup tool. The GUI is identical except for additional options available with the Setup tool.

- **Create Database**. For Oracle and DB2, the **Create Database** option does not create a new physical database. The process only creates a new tablespace in an existing database. Then it creates a database schema. For Oracle, a new user is created before schema creation.

Because SSO is the first component of SOA Systinet that you install, create a database when you install SSO and create new schema in this database when you install the other components.

Important: Before running any SOA Systinet installers, be sure the database is set up as described in Database Setup Overview on page 25.

- **Create Schema (default).** Create tables and indexes in the default schema in an existing database. Select this method if you have access to an existing empty database with the ability to create tables and indexes. This option is suitable when you do not know the administrator's credentials. We assume the administrator has already created a new database/user/tablespace for this option.
- **Drop Database (Setup tool only).** The reverse of creating a database. Details depend on the type of database. Anything you did manually when creating the database, you must undo manually. You need an administrator's credentials.
- **Drop Schema (Setup tool only).** Drops all tables in the database but leaves the empty database.
- **Configure Database (Setup tool only).** Use this option with an existing database. For example, use this method if the database already exists from a previous SOA Systinet installation of the same release number.

After you select the operation, type in the database parameters described in Table 7 on page 29.

Important: Tables for all SOA Systinet components must share the same schema. Therefore, set identical database parameters for each component.

Table 7. Database Setup Parameters

Parameter	Description	Notes
Database Server Address	The hostname or IP address where the database server is accessible.	For example, in the database connection string `jdbc:oracle:thin:@dbhost42:1521:platform`, the hostname is `dbhost42`.
Database Server Port	The connection port for the database.	For example, in the database connection string `jdbc:oracle:thin:@dbhost42:1521:platform`, the port number is `1521`.
Existing Database Name	The name of the database.	For example, in the database connection string `jdbc:oracle:thin:@dbhost42:1521:platform`, the database name is `platform`.
Database Administrator Name	The user name and password of the administrator of the database.	Only required for the **Create Database** and **Drop Database** options.
Database Administrator Password		
Database Tablespace Name	Name of the tablespace you create with the **Create Database** option. For DB2, enter the name of an existing tablespace for the **Create Schema** option.	For both Oracle and DB2, you must type a new tablespace name when you create a database. For DB2, you must type the name of an existing tablespace when you create a schema. When you type a new tablespace name, the tablespace name must not conflict with existing objects in the database.
Tablespace Datafile	The path to the tablespace datafile that is stored on the database host machine.	Only required to create a new database tablespace. Must not conflict with existing objects in the database.

Parameter	Description	Notes
(New) Database User	The name and password of a user who can create tables in his default schema, for the **Create Schema** option. Default user name is "platform" and default password is "changeit." You must confirm a new user's password if creating a database.	When you create a new user, the user's name must not conflict with existing objects in the database.
(New) Database User Password		
Confirm Password		

6 Manual Database Arrangement

This chapter describes how to arrange the database manually. It is useful for enterprise deployment, when the database administrators do not give out permissions for creating schema items (tables, indexes, sequences etc.). For evaluation purposes, a database account with full permissions can be created and the SOA Systinet installers can arrange the database automatically.

The SOA Systinet installer and Setup tool cannot be used to create schema if the database account SOA Systinet uses does not have permission to create schema items. In these cases a database administrator must create schema items on a "power_user" account and grant access to these items to a "common_user" account, which SOA Systinet then uses. In order to perform database operations manually, the database administrator needs to understand how the SOA Systinet installers and Setup tool perform them. This chapter explains how the SOA Systinet tools perform database operations, suggests how to perform the same tasks manually, and describes how to give a "common_user" account access to schemas created on a "power_user" account.

At installation time, the database administrator needs to perform the following tasks:

1. Create a database as described in Setting Up The "power_user" Account on page 31
2. Within the database, create a schema for each SOA Systinet component, described in Creating Schema Manually on page 33
3. If SOA Systinet should use a database with minimal privileges, create a "common_user" account and grant this account privileges to access the created schema, as described in Running SOA Systinet With Minimal Privileges on page 33.

When extensions are applied to an existing installation of SOA Systinet that uses a common_user account, the database administrator needs to perform the tasks described in When Extensions Are Applied on page 37.

Setting Up The "power_user" Account

This section describes how to manually set up a "power_user" account and gives the location of the SOA Systinet installer scripts that create this account.

You only need to set up one power_user. The SOA Systinet administrator needs to provide you with the scripts from any one SOA Systinet component. The scripts with each component are identical. The power_user you create can then be used to create schema resources for all SOA Systinet components.

Setting Up an Oracle power_user

To set up a power_user on Oracle:

1. Create a "power_user" account that can create schema items.
2. Create a tablespace
3. Grant privileges to the power_user for connecting the database and creating tables, indexes, and sequences
4. (Optional) Grant the power_user the privilege to execute "CTXSYS"."CTX_DDL." This is a precondition for using the SOA Systinet Full Text Search feature on the database.

The SOA Systinet installer **Create Database** option for Oracle executes the script `INSTALLATION_JAR#distribution\conf\setup\steps\rdbms_setup\oracle\createdb.sql`. You can set up the power_user manually by running this script with a tool like `sqlplus`..

Setting Up a DB2 power_user

To set up a power_user on DB2:

1. Create a "power_user" account that can create schema items.
2. Create a tablespace
3. Grant CREATETAB, CONNECT and IMPLICIT_SCHEMA privileges to the power_user
4. Grant use of the created tablespace to the power_user

The SOA Systinet installer **Create Database** option for DB2 executes the ANT scripts `INSTALLATION_JAR#distribution\conf\setup\steps\rdbms_setup\db2\installDb2.xml` and `INSTALLATION_JAR#distribution/conf/setup/steps/cfg_db_setup/sql/dbschema_db2_cfg_create.sql`. `rdbms_setup\db2\installDb2.xml` creates the tablespace and and `dbschema_db2_cfg_create.sql` creates the configuration tables. You can perform the same operations in the DB2 Control Center or in an SQL command

line editor. After you create the database, create schema as described in How Schema Are Created on DB2 on page 33.

Creating Schema Manually

Schema can only be created on a "power_user" account, with the rights to create tables, indexes and sequences, where credentials are sensitive. To avoid giving out these credentials, the database administrator can create schema manually instead of using SOA Systinet tools. Manual schema creation only makes sense in scenarios involving two database accounts: a "power_user" account used for installation and upgrades and a "common_user" account with minimal privileges, for everyday database operations. This section explains how to create schema in this scenario.

Schema must be created for each SOA Systinet component.

How Schema Are Created on Oracle

The SOA Systinet installers and Setup tool execute `INSTALLATION_JAR#distribution\conf\setup\steps\rdbms_setup\oracle\schema_core.sql`. You can perform the operations in this script by using an SQL tool. You must create a schema for each component of SOA Systinet you are installing.

How Schema Are Created on DB2

The SOA Systinet installers and Setup tool execute `INSTALLATION_JAR#distribution\conf\setup\steps\rdbms_setup\db2\schema_core.sql`. You can perform the operations in this script by using an SQL tool, but replace "&1" with the name of the tablespace where tables should be created. You must create a schema for each component of SOA Systinet you are installing.

Running SOA Systinet With Minimal Privileges

This section describes how to make schema created in a "power_user" account (see Creating Schema Manually on page 33) accessible to a "common_user" account. To make schema accessible:

- Create a common_user with permission to create alias/synonyms.
- Grant access rights for tables and sequences to common_user.

- Create a synonym/alias for each table and sequence in the common_user's schema.

Configuring "common_user" on DB2

To grant access rights and create aliases for a "common_user" account:

1. Open the DB2 Command Editor. Connect to the database using the "power_user" credentials.
2. Execute the following statement:

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

3. After the results display, click **Fetch More Rows**.
4. Select all results and copy them to the clipboard.
5. In the **Commands** window, paste the clipboard contents.
6. Execute the commands.
7. Execute the following statement:

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

8. After the results display, click **Fetch More Rows**.
9. Select all results and copy them to the clipboard.
10. Open a new instance of DB2 Command Editor. Connect to the database using the "common_user" credentials.
11. In the common_user **Commands** window, paste the clipboard contents.
12. Execute the commands.

Configuring "common_user" on Oracle

To grant access rights and create aliases for a "common_user" account:

1 Save the following SQL statements to the `script.sql` file. These statements set the environment, grant rights and create synonyms. Run with parameters `common_user` and `power_user` with real values.

```
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
```

2 Connect to the database as power_user and process the `script.sql` file. This produces the scripts `grant.sql` and `synonyms.sql`. Then run `grant.sql`.

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

3 As common_user, run `synonyms.sql`.

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

Example 1 on page 36 shows the configuration steps done by an Oracle database administrator on a UNIX system using bash. These steps require sqlplus to be on the path.

Example 1: Database Administrator Configuration Steps for Oracle

```
STEP2: CREATING USERS AND TABLES ON ORACLE
==========================================

export ORACLE_SID=platform
export ORACLE_HOME=/local/oracle/product/10.2/db1

Extract hp-soa-systinet-2.51-decoupled.zip to tmp folder.

CREATING USERS
==============

cd dc/sso
sqlplus "/ as sysdba"
SQL> @createdb.sql power_user_space power_user changeit
     /local/oracle/product/10.2/oradata/platform/power_user_space.dbf
SQL> @createcommon.sql power_user_space common_user changeit
SQL> quit

CREATING TABLES
===============

sqlplus power_user
SQL> @dbschema_oracle_cfg_create.sql
SQL> @schema_core.sql
SQL> quit

cd ../reporting/
sqlplus power_user
SQL> @schema_core.sql
SQL> commit
SQL> quit

cd ../platform/
sqlplus power_user
SQL> @schema_core.sql
SQL> quit

cd ../policymgr/
sqlplus power_user
SQL> @schema_core.sql
SQL> quit

CONFIGURING COMMON_USER
```

```
======================

cd ../sso
sqlplus power_user
SQL> @script.sql power_user common_user
SQL> @grant.sql
SQL> commit;
SQL> quit

sqlplus common_user
SQL> @synonyms.sql
SQL> commit;
SQL> quit
```

When Extensions Are Applied

When extensions are applied to SOA Systinet with a database with minimal privileges, the database administrator must regenerate schema items. The tasks that the database administrator must perform are:

1. Drop the database tables
2. Create new database tables
3. Insert initial data into the database tables
4. Recreate the alias/synonym for the database.

Scripts for dropping and creating tables and for inserting initial data can be provided by the SOA Systinet administrator, from the `PLATFORM_HOME\conf\sql` directory. Also note that the SOA Systinet administrator backs up and restores the database contents using the SOA Systinet Setup tool.

Part III. Deploying SOA Systinet To Application Servers

The components of SOA Systinet are deployed to J2EE application servers. This part is a guide to setting up each supported J2EE server and deploying SOA Systinet to that server.

This part contains the following chapters:

- Installing SOA Systinet on JBoss on page 41. A high-level view of using the SOA Systinet installers to automatically install SOA Systinet to the JBoss server, along with post-installation JBoss setup and launch instructions.
- Deploying SOA Systinet to WebLogic on page 59. Instructions for setting up a WebLogic profile and its clusters and servers to host SOA Systinet. Instructions on deploying SOA Systinet EAR files to the profile's servers.
- Deploying SOA Systinet to WebSphere on page 81. Instructions for setting up a WebSphere domain and its clusters and servers to host SOA Systinet. Instructions on deploying SOA Systinet EAR files to the domain's servers.
- Deploying SOA Systinet to Oracle Application Server on page 101. Instructions for setting up messaging on an existing OAS installation and deploying SOA Systinet to OAS.

7 Installing SOA Systinet on JBoss

Installation to JBoss requires less setup than installation to other J2EE servers. The SOA Systinet installation wizards automate deployment. They set up datasources and JMS on the host JBoss servers and then deploy the SOA Systinet EAR files. The installer also creates a script for setting up the server environment and launching JBoss in simple deployment scenarios. After installation you must still configure the JBoss server, however. Installation to JBoss is described in these sections:

- High-level JBoss Installation Procedure on page 41
- Clustered JBoss on page 42
- Setting Up the JBoss Server on page 48
- Launching SOA Systinet on JBoss on page 57

High-level JBoss Installation Procedure

The SOA Systinet component installers automatically configure and install SOA Systinet on the JBoss application server. You do not need to configure the JBoss server prior to installation. Install the components of SOA Systinet in the following order, regardless of what type of deployment you have designed (see Designing Your Deployment on page 19).

1. Install the SSO service. See Installing the Single Sign-On Service on page 117.
2. Configure the SSO host JBoss server as necessary. See Setting Up the JBoss Server on page 48.
3. Launch the SSO service. See Launching SOA Systinet on JBoss on page 57.
4. Install Reporting Service. See Installing Reporting Service on page 129
5. Configure the Reporting Service host JBoss server as necessary. See Setting Up the JBoss Server on page 48.

6 Launch the Reporting Service JBoss server. See Launching SOA Systinet on JBoss on page 57

7 Install SOA Systinet Platform. See Installing Platform on page 135

8 Configure the SOA Systinet Platform host JBoss server as necessary. See Setting Up the JBoss Server on page 48.

9 Launch SOA Systinet Platform. See Launching SOA Systinet on JBoss on page 57

10 Install Policy Manager. See Installing Policy Manager on page 143

11 Configure the Policy Manager host JBoss server as necessary. See Setting Up the JBoss Server on page 48.

12 Launch SOA Systinet. See Launching SOA Systinet on JBoss on page 57.

Clustered JBoss

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

This section gives instructions for creating a cluster of two computers where each hosts one JBoss instance with all SOA Systinet products. (You can scale up the procedure for more computers.) A JBoss instance with configuration named `node1` is located on the first computer and `node2` on the second one. A load balancer runs on the first computer. Clustered deployment was tested on JBoss-4.0.5.GA.

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 1 on page 43.

Example 1: 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/uriworkermap.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 2 on page 45 is a modified `workers.properties` file.
6. Run the Apache server with the configured load balancer.

Example 2: 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=b136
worker.node1.type=ajp13
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=b137
worker.node2.type=ajp13
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 SSO to JBoss Cluster

Install SSO to a temporary JBoss configuration.

To install SSO when clustering JBoss:

1. Copy the `JBOSS_HOME/server/all` configuration. Name the copy `nodeX`.
2. Launch the SSO service installer (see Installing the Single Sign-On Service on page 117).
3. In the **Endpoint Properties** step, pass the load balancer hostname and endpoints.

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

Creating First Node

After the load balancer has started running and you have installed SSO to a temporary configuration, create the first node that you will 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 the Oracle DB, see Using Oracle DS in Clustered Deployments on page 50. For DB2, see Using DB2 DS in Clustered Deployments on page 52.

3 Copy the `nodeX` datasource, `JBOSS_HOME\server\nodeX\deploy\hpsoasystinet-xa-ds.xml`. Paste it into `JBOSS_HOME\server\node1\deploy\`

4 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\jbossweb-tomcat55.sar\META-INF\jboss-service.xml`

5 Open the file `JBOSS_HOME\server\node1\deploy\jbossweb-tomcat55.sar\server.xml`. Edit it in the next few steps.

6 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.

7 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.

8 Apply the following workaround to disable session replication among clusters:

   a Open the file `JBOSS_HOME\server\node1\deploy\tc5-cluster.sar\META-INF\jboss-service.xml`.

   b Change the value of `buddyReplicationEnabled` from false to true. Change the value of `numBuddies` from 1 to 0.

9 Deploy SSO service to `node1`. Copy `JBOSS_HOME\server\nodeX\deploy\hp-soa-systinet-sso.ear` to `JBOSS_HOME\server\node1\farm\`

10 Run `node1` and try to connect `http://loadBalancerHostname:port/sso/rest/descriptor`. It should return a list of registered entities in SSO server.

Install Other Components

With SSO running on `node1`, install all remaining components. Follow these rules:

- In the **Endpoint Properties** step, pass the load balancer hostname and endpoints.
- In the **Application Server Properties** step, deploy every component to the `nodeX` configuration.
- Install the components in the following order:
  1. Reporting
  2. Platform
  3. Policy Manager

Deploying All Components

After installing all SOA Systinet components to a temporary configuration, deploy them to the cluster nodes.

1 Stop the `node1` server. You can delete its `log`, `tmp`, and `work` directories.

2 Copy the following JMS configuration files from `JBOSS_HOME\server\nodeX\deploy` to `JBOSS_HOME\server\node1\deploy-hasingleton\jms`:

- `queue-ReportingExecutions-service.xml`
- `queue-scheduleTimerQueue-service.xml`
- `queue-taskProcessorQueue-service.xml`
- `queue-Validation-service.xml`

- `topic-taskStopperTopic-service.xml`

3 Create additional cluster nodes.

   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.

4 Copy the following ear files from JBOSS_HOME\server\nodeX\deploy to JBOSS_HOME\server\node1\farm\. They will be distributed to all other cluster nodes when those nodes boot.

   - hp-soa-systinet-platform.ear
   - hp-soa-systinet-policymgr.ear
   - hp-soa-systinet-reporting.ear

5 Launch `node1` on the first computer. When it has successfully started, 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 -c nodeName
      -Dplatform.hajndi.url=jnp://hostname:1100/
      -Dpolicymgr.hajndi.url=jnp://hostname:1100/
      -Djboss.partition.name=clusterName
```

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

Setting Up the JBoss Server

You need to modify the JBoss application server for it to host SOA Systinet, particularly in production environments. These modifications are covered in the following sections:

- JBoss JMS Configuration on page 49. Configure JBoss to use the Oracle or DB2 database instead of the default HSQLDB.
- Modifying the JBoss Run Script on page 53. 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 Datasource MaxPoolSize on page 55. Increase the datasource maximum pool size for production deployments.
- Configuring JBoss When SOA Systinet Uses Non-default Ports on page 56. If you set the SOA Systinet endpoint to use ports other than the default 8080 and 8443, enable these ports on JBoss.

JBoss JMS Configuration

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 Oracle or DB2. Follow one of these procedures for changing the JMS configuration:

- Using Oracle DS in Non-Clustered Deployments on page 49
- Using Oracle DS in Clustered Deployments on page 50
- Using DB2 DS in Non-Clustered Deployments on page 51
- Using DB2 DS in Clustered Deployments on page 52

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 parallel served execution requests (must be less than the number of parallel served users) plus the number of parallel processed executions (~5).

7 Save `oracle-ds.xml`.

8 Delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml`.

9 Copy `JBOSS_HOME/docs/examples/jms/oracle-jdbc2-service.xml` into the folder `JBOSS_HOME/server/default/deploy/jms`

10 In the new copy of `oracle-jdbc2-service.xml`, replace the string `OracleDS` with `DefaultDS`.

11 Save `oracle-jdbc2-service.xml`.

12 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.

13 Open `JBOSS_HOME/server/default/deploy/jbossweb-tomcat55.sar/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using Oracle DS in Clustered Deployments

To set up JBoss JMS to use the Oracle DS in clustered deployments (*node1* in the path refers to a copy of the `all` configuration folder):

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

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 parallel served execution requests (must be less than the number of parallel served users) plus the number of parallel processed executions (~5).

7 Save `oracle-ds.xml`.

8 Delete the file `JBOSS_HOME/server/`*`node1`*`/deploy-hasingleton/jms/hsqldb-jdbc2-service.xml`.

9 Copy `JBOSS_HOME/docs/examples/jms/oracle-jdbc2-service.xml` into the folder `JBOSS_HOME/server/`*`node1`*`/deploy-hasingleton/jms`

10 In the new copy of `oracle-jdbc2-service.xml`, replace the string `OracleDS` with `DefaultDS`.

11 Save `oracle-jdbc2-service.xml`.

12 Open `JBOSS_HOME/server/d`*`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/jbossweb-tomcat55.sar/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Using DB2 DS in Non-Clustered Deployments

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

1 Download a copy of JBoss 4.2.x. You need it for Step 11.

2 Copy the DB2 JDBC drivers `db2jcc.jar` and `db2jcc_license_cu.jar` to `JBOSS_HOME/server/default/lib`.

3 Delete the file `JBOSS_HOME/server/default/deploy/hsqldb-ds.xml` .

4 Copy `JBOSS_HOME/docs/examples/jca/db2-ds.xml` to `JBOSS_HOME/server/default/deploy`.

5 In the new copy of `db2-ds.xml`, edit the `connection-url`, `user-name` and `password` elements to match your local environment.

6 Change the value of the `driver-class` element to `com.ibm.db2.jcc.DB2Driver`.

7 Change the value of the `jndi-name` element from `DB2DS` to `DefaultDS`.

8 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 parallel served execution requests (must be less than the number of parallel served users) plus the number of parallel processed executions (~5).

9 Save `db2-ds.xml`.

10 Delete the file `JBOSS_HOME/server/default/deploy/jms/hsqldb-jdbc2-service.xml`.

11 Copy `JBOSS_HOME/docs/examples/jms/db2-jdbc2-service.xml` into the folder `JBOSS_HOMEserver/default/deploy/jms`

The file `db2-jdbc2-service.xml` has to be taken from a copy of JBoss version 4.2.x as version 4.0.5 does not contain it.

12 In the new copy of `db2-jdbc2-service.xml`, replace the string `DB2DS` with `DefaultDS`.

13 Save `db2-jdbc2-service.xml`.

14 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.

15 Open `JBOSS_HOME/server/default/deploy/jbossweb-tomcat55.sar/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 (*node1* in the path refers to a copy of the `all` configuration folder):

1 Download a copy of JBoss 4.2.x. You need it for Step 11.

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

3 Delete the file `JBOSS_HOME/server/default/`*node1*`/hsqldb-ds.xml` .

4 Copy `JBOSS_HOME/docs/examples/jca/db2-ds.xml` to `JBOSS_HOME/server/`*node1*`/deploy`.

5 In the new copy of `db2-ds.xml`, edit the `connection-url`, `user-name` and `password` elements to match your local environment.

6 Change the value of the `driver-class` element to `com.ibm.db2.jcc.DB2Driver`.

7 Change the value of the `jndi-name` element from `DB2DS` to `DefaultDS`.

8 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 parallel served execution requests (must be less than the number of parallel served users) plus the number of parallel processed executions (~5).

9 Save `db2-ds.xml`.

10 Delete the file `JBOSS_HOME/server/`*`node1`*`/deploy-hasingleton/jms/hsqldb-jdbc2-service.xml`.

11 Copy `JBOSS_HOME/docs/examples/jms/db2-jdbc2-service.xml` into the folder `JBOSS_HOME/server/`*`node1`*`/deploy-hasingleton/jms`

The file `db2-jdbc2-service.xml` has to be taken from a copy of JBoss version 4.2.x as version 4.0.5 does not contain it.

12 In the new copy of `db2-jdbc2-service.xml`, replace the string `DB2DS` with `DefaultDS`.

13 Save `db2-jdbc2-service.xml`.

14 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.

15 Open `JBOSS_HOME/server/`*`node1`*`/deploy/jbossweb-tomcat55.sar/server.xml` and set the `maxThreads` attribute to the maximum number of parallel served users.

Modifying the JBoss Run Script

When you launch SOA Systinet with the `SYSTINET_COMPONENT_HOME/bin/serverstart` script, it sets up the JBoss environment before calling the JBoss run script. No further setup 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:

- JBoss Client Truststore on page 54
- JBoss Memory Allocation on page 55

JBoss Client Truststore

For SSL communication, each JBoss server must access the client truststore of a SOA Systinet component that is deployed to it. If more than one component is deployed to the same JBoss, that JBoss server needs to access only one of the component truststores, because all truststores contain the same CA certificate. However, even if all SOA Systinet components are deployed to the same JBoss, you must still give that JBoss server access to at least one component truststore.

To give a JBoss server access to SOA Systinet client truststores:

1. Open the `JBOSS_HOME/bin/run.bat|run.conf` script in an editor.
2. Insert these lines where JAVA_OPTS is set:

```
-Djavax.net.ssl.trustStore=SYSTINET_COMPONENT_HOME\conf\client.truststore
            -Djavax.net.ssl.trustStorePassword=changeit
```

3. Save and exit the script.

Setting awt.headless

If JBoss will host Reporting or Policy Manager, 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. This is suggested for JBoss servers hosting SSO, Policy Manager and/or Platform servers. It is not necessary for the Reporting host server.

To increase the maximum memory limit to 1.2 GB and set the MaxPermSize to 256m:

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
```

Note: 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.

Setting 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 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 When SOA Systinet Uses Non-default Ports

SOA Systinet by default uses ports 8080 and 8443. If you select a different set of ports during installation, you have to 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 as follows:

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.

Launching SOA Systinet on JBoss

The `bin` directory of every SOA Systinet component contains the scripts `serverstart`, `serverstop` and `env-jboss`. Running `serverstart` calls `env-jboss`, which sets up the JBoss environment. Specifically, `env-jboss` gives JBoss access to the SOA Systinet client truststore and optimizes JBoss memory allocation. Using `serverstart` and `serverstop` is therefore the simplest way of launching and stopping SOA Systinet.

If all SOA Systinet components are installed to the same JBoss, it is only necessary to run one of the component `serverstart` scripts. It does not matter which one.

Important: The JBoss application server attempts to hot-deploy each EAR file after every component installation. Do not rely on this hot-deployment feature, especially if you are hosting multiple J2EE servers on one JBoss installation. Stop and restart any running JBoss nodes after each component installation ends.

In some production environments—where the SOA Systinet components are widely distributed or clustered, when there are applications other than SOA Systinet on the same JBoss, 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 Setting Up the JBoss Server on page 48. The contents of `serverstart` and `serverstop` are also useful guides in this case.

8 Deploying SOA Systinet to WebLogic

Set up a separate WebLogic domain for hosting SOA Systinet. Use the Sun Java vendor for this domain and all its managed servers and/or clusters. You need to configure the domain itself and JDBC and JMS properties for the managed servers and/or clusters in the domain. Before creating the domain, see Designing Your Deployment on page 19. You need to know the number and locations of cluster servers or managed servers you need before you start.

Important: You must have the correct version of WebLogic and the correct updates as given in Supported Platforms on page 17

WebLogic setup is described in the following sections:

- Creating a Domain on page 59
- Setting Up Trust on WebLogic on page 65
- Creating JDBC Resources on page 67
- Creating JMS Resources on page 70
- Creating a Mail Session on page 77
- Deploying SOA Systinet Components on page 77
- Starting and Verifying SOA Systinet on page 78

Creating a Domain

To set up a WebLogic domain for SOA Systinet:

1 Create a new domain using the BEA WebLogic Configuration Wizard.

a **UNIX/Linux on remote server via x server:** On the remote server, set up the DISPLAY property, for example `export DISPLAY=your_workstation:0.0`. On your workstation, add the remote server to your list of trusted hosts by entering `xhost remote_server` . You might need to re-configure the firewall on your workstation to allow the server to access x server on your workstation (allow access to ports 6000:6064/tcp).

b Launch the BEA WebLogic Configuration Wizard. On Windows, you can do this in the Windows **Start** menu. From all operating systems, you can launch the configuration wizard by running `BEA_HOME/weblogic92/common/bin/config.bat|config.sh`.

c Select **Create a New Domain**. Click **Next**. The **Select Domain Source** page opens.

d You can use the default settings for the domain source and click **Next**. The admin username window opens.

e Set admin username and password and click **Next**. The `Configure Server Start Mode and JDK` page opens.

f You may select either development or production mode.

g Select the BEA-provided Sun 1.5.0 SDK. Click **Next**. The **Customize Environment and Services Settings** page opens.

h Select **Yes** and click **Next**.

i You may leave Administrator Server settings at default or change them. Click **Next** when done.

j Create the managed servers required by your deployment design (See Designing Your Deployment on page 19). You need at least one managed server—do not install SOA Systinet to the administration server. Give the servers arbitrary names, such as "server1" or "pmhost2." You can use a proxy for clusters. Click **Next** when done.

k Create the clusters required by your deployment design. Click **Next** when done.

l Assign managed servers to any clusters you created. Click **Next** when done.

m You can create an HTTP proxy server for any clusters you created but this is optional. Click **Next** when done.

n Create a machine in this domain. Click **Next** when done.

o 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** when done.

p Review the new domain and click **Next** when done.

q Set the domain name and location. Click **Create** to create the domain.

The path to the domain is called DOMAIN_HOME in the rest of this chapter.

2 SOA Systinet uses log4j.logging. SOA Systinet provides a copy of `log4j.jar` and puts it on the classpath, but you must create your own log4j configuration file. Save the log4j configuration file to a suitable location in DOMAIN_HOME, with the name `log4j.config`.

Example 1 on page 62 is a log4j configuration you can copy and paste. This configuration instructs log4j to log to both console and a file named `log4j.log`, which log4j creates by default in DOMAIN_HOME. This configuration also strips overly verbose log4j categories down to ERROR and FATAL logs.

Example 1: Log4j Configuration File

```
# console appender
log4j.appender.stderr=org.apache.log4j.ConsoleAppender
log4j.appender.stderr.Target=System.err
log4j.appender.stderr.layout=org.apache.log4j.PatternLayout
log4j.appender.stderr.layout.ConversionPattern=%p: %c{2} - %m%n

# file appender
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.maxFileSize=100KB
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

# save all logs in stderr and file
log4j.rootLogger=INFO,stderr,file

# limit categories that are too verbose
log4j.category.org.apache.xml.security=ERROR,stderr
log4j.additivity.org.apache.xml.security=true
log4j.category.org.apache.beehive=ERROR,stderr,file
log4j.additivity.org.apache.beehive=true

#configure hibernate (prevents certain spurious error messages)
log4j.category.org.hibernate=OFF,stderr,file
log4j.additivity.org.hibernate=true
```

3 Open `DOMAIN_HOME/config/config.xml`. Add `<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>` inside and at the end of `<security-configuration>`. (If `<enforce-valid-basic-auth-credentials>` is already present, set its value to `false`.) SOA Systinet uses its own authentication headers.

Caution: The Administration Server *must not be running* when you modify `config.xml`. Otherwise, the Administration Server can overwrite your changes.

4 Start Node Manager. On Windows, start Node Manager from the **Tools** folder in the **BEA Products** Start menu. On UNIX systems, run **WEBLOGIC_HOME/weblogic92/server/bin/startNodeManager.sh**

5 Start the Administration Server and open the Administration Console. On UNIX systems, run **DOMAIN_HOME/startWebLogic.sh**.

6 Use the Sun Java vendor for this domain and all its managed servers whether standalone or clustered. For each server, in the Administration Console go to **Environment**->**Servers**-> ***server_name*** ->**Configuration**->**Server Start**. In the **Java Vendor** field, type `Sun`. In the **Java Home** field, type `BEA_HOME\jdk150_10`. Alternatively, you can set JAVA_VENDOR programatically or in the `setDomainEnv` and `startManagedWebLogic` scripts. Be certain to set the vendor to Sun for every managed server and/or cluster server.

Caution: SOA Systinet does not support JRockit. All managed servers and/or cluster servers hosting SOA Systinet must use Sun Java.

7 For every server in the domain, whether standalone or clustered, add Java arguments to do the following:

- Allocate memory.
- Specify the location of the `log4j.config` file.
- For the Policy Manager host, set AWT headless to "true."

To add Java arguments:

a Go to **Environment**->**Servers**-> ***server_name*** ->**Configuration**->**Server Start**. Type Java arguments into the **Arguments** field.

b To allocate memory, type `-Xms128m -Xmx1300m -XX:MaxPermSize=128m` into the **Arguments** field. (If all SOA Systinet components are installed to one managed server, set MaxPermSize to 384m instead of 128m.) -Xmx is the maximum heap size and can be set higher depending on the capabilities of the computer, but it must not be lower than 1300. See Sizing on page 21 for details.

c Also in the **Arguments** field, type the path to the `log4j.config` file. Use the absolute path: `-Dlog4j.configuration=file:[url-style absolute path to log4j.config file]` . For example, the

argument may be `file:\\\c:/bea/user_projects/systinet2/bin/log4j.config` or `file:/usr/bea/user_projects/systinet2/bin/log4j.config`.

d For the WebLogic server(s) hosting Policy Manager, also add this argument: `-Djava.awt.headless=true`.

e Save your changes.

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

```
startMyNode.bat (Windows):
set DOMAIN_HOME=c:\your\weblogic\domain\home

set JAVA_OPTIONS="-Djava.awt.headless=true
            -Dlog4j.configuration=file:%DOMAIN_HOME%/log4j.config
            -Djavax.net.ssl.trustStore=%DOMAIN_HOME%/trust.jks
            -Djavax.net.ssl.trustStorePassword=changeit"

set USER_MEM_ARGS="-Xms512m -Xmx1300m -XX:MaxPermSize=384m"

%DOMAIN_HOME%/bin/startManagedWebLogic [nodename]

startMyNode.sh (Linux):
export $DOMAIN_HOME=/your/weblogic/domain/home

export JAVA_OPTIONS="-Djava.awt.headless=true
      -Dlog4j.configuration=file:$DOMAIN_HOME/log4j.config
      -Djavax.net.ssl.trustStore=$DOMAIN_HOME/trust.jks
      -Djavax.net.ssl.trustStorePassword=changeit"

export USER_MEM_ARGS="-Xms512m -Xmx1300m -XX:MaxPermSize=384m"

. ./bin/startManagedWebLogic.sh [nodename] http://hostname:7303
```

8 Give each server and cluster unique port numbers [**Environment**->**Servers**-> ***server_or_cluster_name*** ->**Configuration**->**General**, fields **Listen Port** and **SSL Listen Port**]. Record these port numbers because you need them when you run the SOA Systinet installation wizards.

9 Set up a load balancer for each cluster. Set up a work manager for each load balancer. To improve performance, set the minimum number of threads in each work manager to at least 2 times the number of expected concurrent users.

Setting Up Trust on WebLogic

If SOA Systinet uses SSL communication, set up trust, even if all components are installed to the same managed server. If SOA Systinet will be synchronized with HP SOA Systinet Registry, the SOA Systinet Platform server has to trust the HP SOA Systinet Registry server. Trust requires a certificate from every server, which you can obtain in one of the following ways:

- From a certification authority such as Verisign
- From your corporate authority
- By creating a custom certificate yourself (described in Step 1 of the following procedure. All other steps of the procedure apply in all cases.)

To set up server trust:

1 Create a keystore with a certificate for every server. You can use the Java `keytool` utility with this command: `keytool -keyalg rsa -genkey -alias server1 -keystore .\server1id.jks`. This command creates a certificate with the alias `server1` in the keystore `.\server1id.jks`. You will be asked to enter a new password for the keystore and a new password for the key pair. Use the same password. You will need it in Step 4.

Important: In the course of creating the certificate, you will be prompted to provide a common name ("your first and last name") for the certificate. Type in the hostname for the server in exactly the same format you used when installing the server.

2 Export the certificate. You can use a `keytool` command such as `keytool -export -alias server1 -keystore .\server1id.jks -file server1.crt`. This exports the certificate with alias server1 to a file called server1.crt. You will be asked for the certificate password you created in Step 1. Store the certificate in a file you specify with the `-file` option.

3 Create either a common trust store or a trust store for every server. If you create a common trust store, it must contain every certificate you created in Step 1 and exported in Step 2. If you create a trust store for every server, each trust store must contain the certificates of the servers with which its server communicates. The trust store is created automatically the first time you import a certificate to it. With the `keytool` utility, enter the command `keytool -import -alias server1 -keystore .\trust.jks -file server1.crt`.

If you are synchronizing SOA Systinet with HP SOA Systinet Registry you need to create a trust store for the SOA Systinet Platform server or load balancer even if you are not using SSL communication between SOA Systinet components.

4 Add the keystore and truststore to each server's configuration.

a Go to **Environment**->**Servers**. The list of servers opens.

b Click the server's name. The server's details page opens in the **Configuration: General** tab.

c Open the **Configuration: Keystores** tab.

d In the **Keystores** drop-down field, select **Custom Identity and Custom Trust**

e In the **Identity** section, type the path, type and password of the keystore you created for the server in Step 1.

f In the **Trust** section, type the path and password of the trust store you created in Step 3. (Leave type as JKS.) This can be a common trust store for all servers or a specific trust store only for this server.

g Click **Save**.

h Open the **Configuration:SSL** tab.

i In the **Identity and Trust Locations** drop-down box, leave the default **Keystores** selected.

j In the **Identity** section, type the alias and password of the key pair you created for the server in Step 1.

k Click **Save** and activate your changes.

5 Add each server's truststore to the runtime environment of the server. How you do this depends on how you start the server. (Compare to Step 7 in Creating a Domain on page 59.)

For servers started via Node Manager:

a Go to **Environment**->**Servers**. The list of servers opens.

b Click the server's name. The server's details page opens in the **Configuration: General** tab.

c Click open the **Configuration: Server Start** tab.

d In the **Arguments** field, type `-Djavax.net.ssl.trustStore=[TRUSTSTORE_PATH] -Djavax.net.ssl.trustStorePassword=[TRUSTSTORE_PASSWORD]`.

e Save and activate changes.

For servers started via script or on the command line: Add to the JAVA_OPTIONS variable `-Djavax.net.ssl.trustStore=[TRUSTSTORE_PATH] -Djavax.net.ssl.trustStorePassword=[TRUSTSTORE_PASSWORD]`.

Creating JDBC Resources

SOA Systinet components require two JDBC datasources, an XA-enabled datasource and a non-XA-enabled datasource. These datasources handle all traffic between the SOA Systinet components on WebLogic and the database server. Every 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.

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 `SOA Systinet DS`.

3 Give the datasource the JNDI name `hpsoasystinetDS`.

Important: JNDI names must be exact.

4 From the **Database** dropdown list, select the same database type that you use for SOA Systinet components.

5 From the **Database Driver** dropdown list, select an XA-supporting JDBC database driver for the database type. If you are using Oracle, select the Oracle "thin" XA driver. Click **Next** and open the **Transaction Options** page.

6 Click **Next** again. The **Connection Properties** page opens.

7 In the **Connection Properties** page, type the same database parameters you use for SOA Systinet.

8 Target all servers or clusters hosting SOA Systinet.

9 Click **Finish**. You return to the **Summary of JDBC Datasources** page. **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. Its details page opens 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. Click **Save** when done but *do not activate changes*. Remain in the **Configuration:Connection Pool** tab.

13 Increase the **Initial Capacity** to the number of expected concurrent users. Click **Save** when done but *do not activate changes*.

14 Open the **Configuration:Transactions** tab. Select **Use XA Datasource Interface**. Click **Save** when done but *do not activate changes*.

15 Navigate out of the datasource's details page, for example to the **Summary of JDBC Datasources** page. Click **Activate Changes**.

Note: If you do not navigate out of the datasource's details page before you save changes you made to the datasource, you raise a **JDBCSystemResourceMBean cannot be null** exception. This exception is harmless, because the changes to the datasource are activated anyway, but it is better not to raise the exception.

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

- Increase count of connections in the datasource.
- Increase timeout for acquiring connection: increase value of Connection Reserve Timeout [Data Sources -> hpsoasystinetDS -> Connection Pool -> Advanced]. (Default is 10 seconds). Setting 0 (infinite waiting for connection) is not recommended beacuse 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** 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`.

Important: JNDI names must be exact.

4 Set it to use the same database type that you use for SOA Systinet components.

5 From the **Database Driver** dropdown menu, select a non-XA-supporting JDBC database driver for the database type. If using Oracle, select the Oracle "thin" non-XA driver. Click **Next** and open the **Transaction Options** page.

6 Select **Supports Global Transactions**. Click **Next**. The **Connection Properties** page opens.

7 In the **Connection Properties** page, type the same database parameters you use for SOA Systinet.

8 Target all servers or clusters hosting SOA Systinet.

9 Finish and activate changes.

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 for every migratable cluster server and server:

1 Navigate to **Services**->**Persistent Stores**.

2 Click **Lock and Edit**.

3 From the **New** drop-down menu, select **Create a JDBC Store**. The **Create JDBC Store** wizard opens.

4 Give the persistent store a unique, arbitrary name, such as *`SERVER_NAME`* `Store`.

5 In the **Datasource** drop-down field, select the non-XA-enabled datasource you created.

6 Give the persistent store a unique prefix, so the stores do not all try to use the same table.

7 Finish and save changes.

8 Repeat the procedure for every migratable cluster server and/or standalone managed server.

At any point after you complete JDBC resource creation you can create the Single Sign-on service EAR file as described in Installing the Single Sign-On Service on page 117.

Creating JMS Resources

You must set up JMS messaging resources before running any deployed SOA Systinet component other than the SSO service. You also need to create a mail session, although you do not have to give it real values.

Creating JMS Servers

Create a JMS server for each migratable cluster server and each standalone server, except for servers only hosting SSO.

To create each JMS server:

1. Open the JMS Servers page, **Services**->**Messaging**->**JMS Servers**
2. Click **Lock and Edit** and **New**. The **Create a New JMS Server** wizard opens 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 Server`, indicating which managed server or migratable cluster server you will target it to.
4. From the **Persistent Store** drop-down field, select the persistent store of the managed server or migratable cluster server to which you will target this JMS server. Then click **Next**. The **Select Targets** page opens.
5. From the **Target** drop-down field, select the standalone managed server or *migratable* cluster server corresponding to the persistent store you selected in Step 4. Click **Finish** and save your changes.

Creating JMS Modules

Create a JMS module for each of the following SOA Systinet components:

- Reporting Service
- SOA Systinet Platform
- Policy Manager

Creating a JMS Module for Reporting Service

To create a JMS module for Reporting Service:

1. Open the JMS Modules page, **Services**->**Messaging**->**JMS Modules**.
2. Click **Lock and Edit** and **New**. The **Create a New JMS Module** wizard opens.

3. Give the JMS module a unique, arbitrary descriptive name, such as `RF Module`. You may give it any descriptor file name and location, or leave those fields blank for default. Click **Next**.
4. Target the standalone server or the cluster hosting Reporting Service.
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 a sending connection factory for the JMS module.
   a. Click **New** in the **Summary of Resources** table. The **Create a New JMS System Module Resource** wizard opens.
   b. Select to create a **Connection Factory** and click **Next**. The **Create a New Connection Factory** wizard opens.
   c. Give the connection factory a unique, arbitrary descriptive name, such as `Reporting Sender Connection Factory`.
   d. Give the connection factory the JNDI name `jms/ReportingSenderConnectionFactory`.

      Important: JNDI names must be exact.

   e. Use the default targeting, which selects the parent module's target.
   f. Finish and activate changes.
   g. Edit the connection factory and select **XA Connection Factory Enabled** in the **Configuration:Transactions** tab.
   h. If the reporting service host is a cluster, in the **Configuration:Load Balancing** tab.
   i. Save and activate changes.
   j. Return to the JMS module details page.

7 Create a receiving connection factory. Follow the same procedure as for creating a sending connection factory in Step 6, but give the receiving connection factory the JNDI name `jms/ReportingReceiverConnectionFactory`. It is important to enter the JNDI name exactly.

8 If the Reporting Service host is a managed server and not a cluster:

   a If you are not already there, navigate to the Reporting Service JMS module details page.

   b Open the **Subdeployments** tab and create a subdeployment for the JMS module.

   c Set the subdeployment's target as the Reporting Service host managed server's JMS server.

   d Return to the JMS module's details page. Click **New** in the **Summary of Resources** table. The **Create a New JMS System Module Resource** wizard opens.

   e Select to create a **Queue** resource and click **Next**. The **Create a New Queue** wizard opens.

   f Give the queue the JNDI name `queue/ReportingExecutions` and configure it to use the subdeployment. Use the default targeting, which is the same targeting as the subdeployment's.

9 If the Reporting Service host is a cluster, return to the JMS module's details page and create a distributed queue resource with the JNDI name `queue/ReportingExecutions`. Target the distributed queue on the cluster.

Creating a JMS Module for SOA Systinet Platform

To create a JMS module for Platform:

1 Open the JMS Modules page, **Services**->**Messaging**->**JMS Modules**.

2 Click **Lock and Edit** and **New**. The **Create a New JMS Module** wizard opens.

3 Give the JMS module a unique, arbitrary descriptive name, such as `SOA Platform Module`. Click **Next**.

4 Target the standalone server or the cluster hosting Platform.

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 a connection factory for the JMS module.

a Click **New** in the Resources table. A page opens where you select the type of resource to create.

b Select **Connection Factory** and click **Next**. The **Create a New Connection Factory** wizard opens.

c Give the connection factory a unique, arbitrary descriptive name, such as `SOA Platform Connection Factory`.

d Give the connection factory the JNDI name `/ConnectionFactory`.

Important: JNDI names must be exact.

e Use the default targeting, which selects the parent module's target.

f Finish and activate changes.

g Open the connection factory's details page. Open the **Configuration:Transactions** tab and select **XA Connection Factory Enabled** .

h If the SOA Systinet Platform host is a cluster, edit the connection factory and disable server affinity in Configuration>Load Balancing.

i Save and activate changes.

j Return to the Platform JMS module details page.

7 If the Platform host is a managed server and not a cluster:

a Open the **Subdeployments** tab of the JMS module's details page and create a subdeployment for the JMS module.

b Set the subdeployment's target as the Platform host managed server's JMS server.

c Return to the JMS module's details page. Open the **Configuration** tab and create the following resources for the JMS module, configuring each of them to use the module's subdeployment. Use the default targeting of each resource, which is the same targeting as the subdeployment's.

Resource type	JNDI name
Queue	`queue/scheduleTimerQueue`
Queue	`queue/taskProcessorQueue`
Topic	`topic/taskStopperTopic`

8 If the Platform host is a cluster, create the following resources for the JMS module, configuring each of them to target the cluster:

Resource type	JNDI name
Distributed queue	`queue/scheduleTimerQueue`
Distributed queue	`queue/taskProcessorQueue`
Distributed topic	`topic/taskStopperTopic`

Creating a JMS Module for Policy Manager

To create a JMS module for Policy Manager:

1 Open the JMS Modules page, **Services**->**Messaging**->**JMS Modules**.

2 Click **Lock and Edit** and **New**. The **Create a New JMS Module** wizard opens.

3 Give the JMS module a unique, arbitrary descriptive name, such as `SOA PM Module`. Click **Next**.

4 Target the standalone server or the cluster hosting Policy Manager.

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 a connection factory for the JMS module.

a Click **New** in the Resources table. A page opens where you select the type of resource to create.

b Select **Connection Factory** and click **Next**. The **Create a New Connection Factory** wizard opens.

c Give the connection factory a unique, arbitrary descriptive name, such as `PM Connection Factory`.

d Give the connection factory the JNDI name `jms/PMConnectionFactory`.

Important: JNDI names must be exact.

e Use the default targeting, which selects the parent module's target.

f Finish and activate changes.

g Open the connection factory's details page. Open the **Configuration:Transactions** tab and select **XA Connection Factory Enabled**. Return to the connection factory's details page.

h If the Policy Manager host is a cluster, open the **Configuration:Load Balancing** tab and disable server affinity.

i Save and activate changes.

j Return to the JMS module details page.

7 If the Policy Manager host is a managed server and not a cluster:

a Open the **Subdeployments** tab of the JMS module's details page and create a subdeployment for the JMS module.

b Set the subdeployment's target as the Policy Manager host managed server's JMS server.

c Return to the JMS module's details page.

d Create a queue resource with the JNDI name `queue/Validation` and configure it to use the subdeployment. Use the default targeting, which is the same targeting as the subdeployment's.

8 If the Policy Manager host is a cluster, return to the JMS module's details page and create a distributed queue resource with the JNDI name `queue/Validation`. Target the distributed queue on the cluster.

Creating a Mail Session

SOA Systinet requires a mail session even if you will not use it.

To create a mail session:

1 Open the **Mail Sessions** page, **Services**->**Mail Sessions**.

2 Click **Lock and Edit** and then **New**. The **Create a New Mail Session** page opens.

3 Give the mail session a unique, arbitrary descriptive name, such as `SOA Systinet Mail`, and click **Finish**.

4 Open the mail session's details page.

5 Give the mail session the JNDI name `/Mail`

6 Type in any mail properties and click **Save**.

7 Open the **Targets** tab. Target the managed server or cluster hosting Platform and click **Save**.

Deploying SOA Systinet Components

Use the SOA Systinet installers to create deployable EAR files. Deploy and start the SSO Service before creating other EAR files.

To deploy SOA Systinet:

1 Create the SSO service EAR file as described in Installing the Single Sign-On Service on page 117.

2 Start all managed servers.

3 In the WebLogic Administration Console, open the **Deployments** page.

4 Click **Lock and Edit** and then **Install**. A tree view of the file system opens.

5 Browse to `SSO_HOME/deploy/hp-soa-systinet-sso.ear`. Select it and click **Next**.

6 Select **Install this deployment as an application** and click **Next**.

7 Select the managed server or cluster you want to host SSO service and click **Next**.

8 You can leave all optional settings at default, or change them according to your deployment policy. Click **Finish**. WebLogic creates the **hp-soa-systinet-sso** deployment.

9 Activate changes and start the **hp-soa-systinet-sso** deployment. It must be running for the other SOA Systinet installers to function.

10 Verify that the SSO deployment is running. Try to view its deployment descriptor in a browser window, at *hostname:port*`/sso/rest/descriptor`.

11 Run the other SOA Systinet installers as described in Part IV, "Installing SOA Systinet Components". This creates the following EAR files:

- REPORTING_HOME/deploy/hp-soa-systinet-reporting.ear
- PLATFORM_HOME/deploy/hp-soa-systinet-platform.ear
- POLICYMGR_HOME/deploy/hp-soa-systinet-policymgr.ear

12 Deploy all undeployed EAR files, following the same procedure you used to deploy the SSO service EAR. The location and names of each EAR file differ for each component as listed in Step 11.

Starting and Verifying SOA Systinet

After you configure WebLogic and deploy SOA Systinet, verify that your deployment is successful. The easiest way to do this is to restart all managed servers, which will automatically launch the SOA Systinet deployments when they are restarted.

To verify SOA Systinet:

1 Restart all managed servers.

2 In the WebLogic Administration Console, navigate to **Deployments** and check that all SOA Systinet deployments are in state **Active**. Try to manually start any that are not active.

3 Verify that the SOA Systinet deployments are running. Try to view their deployment descriptors in a browser window, at these addresses:

- *hostname:port*/reporting/
- *hostname:port*/soa/systinet/platform/rest/
- *hostname:port*/policymgr/rest/service/system/product-information/

9 Deploying SOA Systinet to WebSphere

Create a separate WebSphere profile to host SOA Systinet. Set up JDBC, JMS and other messaging resources on this profile and its application servers and clusters. Use the SOA Systinet component installers to create EAR files which you then deploy using WebSphere's deployment tools.

Important: Before setting up SOA Systinet on WebSphere, make certain the database administrator configures Oracle to support SOA Systinet on WebSphere. See Oracle with WebSphere on page 26.

Important: You must have the correct version of WebSphere *and the correct fixpack* as listed in Supported Platforms on page 17.

The procedures for installing SOA Systinet components on WebSphere are described in the following sections:

- Creating a Profile on page 82
- Setting Up Trust on WebSphere on page 82
- Creating a Mail Session on page 83
- Creating JDBC Resources on page 83
- Deploying the SSO Service and Creating EAR Files on page 87
- Setting Up log4j Logging, Memory Allocation and Java Properties on page 89
- Setting Up JMS for the Reporting Service on page 92
- Setting Up JMS for Platform on page 95

- Setting Up JMS for Policy Manager on page 97
- Deploying the Remaining EAR Files on page 99

Creating a Profile

Create a clean WebSphere profile with the "Cell" environment. This profile is stored in `WEBSPHERE_HOME/AppServer/profiles/PROFILE_NAME`. In the rest of this chapter, this location is called WS_PROFILE_HOME.

Create application servers and/or clusters in the new profile as required by your deployment design (See Designing Your Deployment on page 19).

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. See the WebSphere Help.

Setting Up Trust on WebSphere

If the components of SOA Systinet are installed on servers with different host:port pairs and will communicate via SSL, trust has to be established between these servers. If SOA Systinet will be synchronized with HP SOA Systinet Registry, the SOA Systinet Platform server has to trust the HP SOA Systinet Registry server.

To set up server trust:

1. Use the IKEYMAN tool or the WebSphere Administration Console to generate a CMS-type key database and a self-signed certificate for every unique host:port standalone server and every IBM HTTP Server (IHS) being used as a load balancer.
2. If you need to synchronize SOA Systinet with a remote HP SOA Systinet Registry server, export that server's certificate and import it into the keystore of the SOA Systinet Platform host server.
3. If using an IHS for a load balancer, enable SSL on the IHS. Open `WEBSPHERE_HOME\IHS\conf\httpd.conf` and add the following directive at the end of the directives that define modules:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
    Listen 0.0.0.0:443
    <VirtualHost *:443>
         SSLEnable
    </VirtualHost>
```

```
</IfModule>
SSLDisable
          KeyFile "WEBPSHERE_HOME/IHS/key.kdb"
```

4 In the WebSphere Deployment Manager Administration Console, open **Security**->**SSL Certificate and Key Management**->**Key Stores and Certificates**.

5 Select a pair of SOA Systinet server keystores and click **Exchange signers...**.

6 A window opens in which you can set the trusted certificates of one store to be the signers of the other. Swap all certificates and signers and click **OK**.

7 Repeat Step 5-Step 6 until all servers trust each other.

Creating a Mail Session

Create a mail session in the WebSphere Deployment Manager's Administration Console [**Resources**->**Mail**->**Mail Sessions**]. Select your cell in the **Scope** drop-down field and click **New**. A page appears in which you specify the mail session's properties as follows:

- A unique, arbitrary descriptive name, such as `SOA Systinet Mail`
- The JNDI name `/Mail`

Important: JNDI names must be exact!

- Connection settings as per company email setup
- 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.

Create JDBC resources in this order:

1. Create an XA JDBC provider. [84]
2. Create a non-XA JDBC provider. [85]
3. Create an XA JDBC datasource. [85]
4. Create a non-XA JDBC datasource. [87]

To create a JDBC provider for the XA datasource:

1. In the WebSphere Administration Console, navigate to **Resources**->**JDBC**->**JDBC Providers**.
2. In the **Scope** drop-down field, select your cell.
3. Click **New**.
4. Select database type.
5. Under **Provider**, select the driver for your database type. For Oracle, if there is more than one driver available, select a "Thin" driver. For DB2, if there is more than one driver available, select the DB2 Universal driver.
6. For the implementation type, select **XA data source**. The **Name** field automatically fills in with the driver name followed by **(XA)**.
7. For the value of the variable `${driver_name_PATH}`, type in the location of `ojdbc14.jar` (Oracle) or `db2jcc.jar` and `db2jcc_license_cu.jar` (DB2). The DB2 driver files are in `IBM_HOME/SQLLIB/java` by default.
8. Click **Finish** and save your changes.

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

- Increase count of connections in the datasource. Increase the value "Maximum connections" property in [Resources -> JDBC -> Datasources -> HP SOA Systinet DS on Oracle -> Connection pool properties] (or specify 0 for no connection count limit)

- Increase timeout for acquiring connection. Increase value "Connection Timeout" [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 JDBC provider for the non-XA datasource:

Perform the same procedure as for creating a JDBC provider for the XA datasource, except you select the implementation type **Connection pool data source** and the automatically generated name does not end in **(XA)**.

After you create JDBC providers, create JDBC datasources.

To create an XA-enabled JDBC datasource:

1. In the WebSphere Administration Console, navigate to **Resources**->**JDBC**->**Data Sources**.
2. Select your cell as scope from the drop-down field.
3. Click **New**.
4. Give the datasource a unique, arbitrary descriptive name, such as `HP SOA Systinet DS`.
5. Give the datasource the JNDI name `hpsoasystinetDS`. You must type this name exactly.
6. Click **Next**. The **Select JDBC Provider** page opens.
7. Select to use an existing JDBC provider. Select the XA JDBC provider you created from the drop-down field. Click **Next**. The database properties page opens.
8. The database properties you enter depend on the type of database you are using.

   For Oracle:

   a. Type the full URL of the database you plan to use for SOA Systinet, such as `jdbc:oracle:thin:@dbsrv5:1521:systinet`

   b. From the drop-down field, select the data store helper class name for your version of the database.

c Unselect **Use this datasource for CMP**.

d Finish the wizard and save your changes.

For DB2:

a Type the database name, such as `platform`. Ask your database administrator if you do not know what this name is.

b From the **Driver type** drop-down field, select driver type "4."

c Type the server name.

d Type the port number if it differs from the default `50000`.

e Unselect **Use this datasource for CMP**.

f Finish the wizard and save your changes.

9 Open the newly created datasource and create an authentication alias.

a Click **JAAS - J2C authentication data**. A list of authentication aliases opens.

b Click **New**.

c Give an arbitrary string value for the alias.

d For credentials, type the user name and password for the database you use with SOA Systinet.

e Finish and save your changes.

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

a In the **Component-managed authentication alias** drop-down field, select the authentication alias you created in Step 9.

b Under **Authentication alias for XA recovery** select **Use Component-manager authentication alias**.

c Under **Container-managed authentication** go to the **Mapping configuration alias** drop-down field and select **DefaultPrincipalMapping**.

d Click **OK** and save your changes.

11 Reopen the datasource and increase its connection pool size.

a Under **Additional Properties**, click **Connection pool properties**. The **Connection Pool** page opens.

b In the **Maximum connections** field, type a number equal to 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 connections to 100.

c In the **Minimum connections** field, type a number equal to the number of expected concurrent users.

d Click **OK** and save changes.

12 Click **Test connection** to be sure that you configured your datasource correctly.

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

Perform the same procedure as for creating an XA-enabled datasource, with these differences:

- Give the non-XA datasource the JNDI name `jms-hpsoasystinetDS`. The JNDI name must be exact.
- 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.

Deploying the SSO Service and Creating EAR Files

Use the SOA Systinet installation wizards to create EAR files and the WebSphere Administration Console to deploy them. The Single Sign-on (SSO) service must be deployed and running when you create the other EAR files.

Tip: Note the port numbers of the target managed servers before you run the SOA Systinet installers. To find the port numbers of a server, navigate to **Servers**->**Application Servers**, click the server's name to open its details page and in that page click **Ports**. The port numbers are labelled **WC defaulthost** and **WC defaulthost_secure**.

To deploy SOA Systinet on WebSphere:

1. Run the SOA Systinet SSO service installer to create the SSO EAR file. See Installing the Single Sign-On Service on page 117.
2. Start the Node Agent for your WebSphere profile. Run `WS_PROFILE_HOME/bin/startNode`.
3. In the WebSphere Administration Console, start the application server or cluster that will host the SSO service.
4. Launch the enterprise application installation wizard. In the Administration Console, navigate to **Applications**->**Install New Application**. The **Preparing for application installation** page opens.
5. Type or browse the path to the SSO installation JAR file.
6. Select **Prompt me only when additional information is required**.
7. Click **Next**. The **Install new application** page opens.
8. Select the following (could already be selected by default):
   - **Precompile Java Server Pages files**
   - **Distribute application**
   - **Create MBeans for resources**
9. Leave all other options at default settings. Click **Next**. Deployment begins.
10. Navigate to **Applications**->**Enterprise Applications** Click on the name of the deployed SSO Service application: **HP SOA Systinet SSO**. The details page opens for **HP SOA Systinet SSO**.
11. Click **Class loading and update detection**. The class loading page opens.

12 In the **Class loader order** drop-down field, select **Classes loaded with application class loader first**.

13 Type "0" in the **Polling interval for updated files** field.

14 Click **OK**.

15 Navigate back to the HP SOA Systinet SSO application details page.

16 Click **Manage Modules**.

17 Click the **Systinet SSO Service** module. Its details page opens.

18 In the **Class loader order** drop-down field, select **Classes loaded with application class loader first**.

19 Click **OK** and save changes.

20 Start the HP SOA Systinet SSO application.

21 Verify that the SSO application is running correctly. Try to view its deployment descriptor in a browser window, at `hostname:port/sso/rest/descriptor`.

22 Use the SOA Systinet installers to create the EAR files for the remaining components of SOA Systinet. Do not install these EAR files until you set up log4j and JMS, as described in subsequent sections of this chapter. After you set up JMS, restart all servers before deploying EAR files.

Setting Up log4j Logging, Memory Allocation and Java Properties

SOA Systinet relies on log4j logging. You must set up log4j for SOA Systinet components to function. It is convenient to set up memory allocation and Java properties at the same time, as you do it in the same console page where you add `log4j.jar` to the classpath.

To set up log4j logging and other properties:

1 Create a `log4j.properties` file and save it in `WS_DOMAIN_HOME/properties`. You can copy and paste the properties file in Example 1 on page 90.

Example 1: Sample log4j.properties File

```
log4j.appender.stderr=org.apache.log4j.ConsoleAppender
log4j.appender.stderr.Target=System.err
log4j.appender.stderr.layout=org.apache.log4j.PatternLayout
log4j.appender.stderr.layout.ConversionPattern=%p: %c{2} - %m%n

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

log4j.category.org.hibernate=OFF,stderr,file
log4j.additivity.org.hibernate=true

# put all logs to console and a log file
log4j.rootLogger=INFO,file

# limit categories that are too verbose
log4j.category.org.apache.xml.security=ERROR,file,stderr
log4j.additivity.org.apache.xml.security=true
```

2 Copy `PLATFORM_HOME/deploy/websphere/jars/log4j.jar` to the profile of *every* WebSphere server hosting a SOA Systinet component, in `WS_DOMAIN_HOME/jars`. (Create this directory if it does not exist.)

3 Add `log4j.jar` to the classpath of every server, clustered or standalone. Repeat the following steps for *every* WebSphere server hosting SOA Systinet:

a In the server's details page, navigate to **Server Infrastructure**->**Java and Process Management**->**Process Definition**. The **Process Definition** page opens.

b Navigate to **Additional Properties**->**Java Virtual Machine**. The **Java Virtual Machine** page opens.

c In the **Classpath** field, type in the full path to the `log4j.jar` file you copied in Step 2, `WS_DOMAIN_HOME/jars/log4j.jar`.

Keep the **Java Virtual Machine** page open.

d Allocate memory. Set the **Initial Heap** to 1000 and the **Max Heap** to 1500. You can set the maximum heap size higher than this if your system supports it.

   Keep the **Java Virtual Machine** page open.

e If the server will host Reporting or Policy Manager, enter `-Djava.awt.headless=true` in the **Generic JVM arguments** field.

f Click **OK** and save changes.

g Repeat this process for every server—standalone or cluster member—hosting a component of SOA Systinet.

Creating a Messaging Bus

In WebSphere, JMS communication and the persistant storage of that communication are handled via a bus. All SOA Systinet components can share a single bus.

To create a messaging bus:

1 Navigate to **Service Integration**->**Buses**. The **Buses** page opens.

2 Click **New**. The **Create a new bus** wizard opens.

3 Give the bus a unique, arbitrary descriptive name, such as `SOABus`.

4 Unselect **Bus Security**, as it is not necessary, and click **Next**.

5 Finish and save changes. The **Buses** page reopens.

6 Click the name of the bus you created. Its details page opens.

7 Click **Bus members**. The **Bus members** page opens.

8 Click **Add**. The **Add a New Bus Member** wizard opens.

9 Select a standalone server or cluster that will host SOA Systinet components. Click **Next**. The **Select type of message store** page opens.

10 Select **Data store** for the type of message store. Click **Next**.

11 Enter the message store properties for the bus. (You can use the existing `jms-hpsoasystinetDS` datasource but you might prefer to use a different data source for performance reasons.) If you use an existing datasource, in the **Schema name** field, type the database user name. Select **Create tables**. Click **Next**.

12 Review your selected options and click **Finish**. You return to the **Bus members** page.

13 Repeat Step 8-Step 12 for every standalone server and cluster that will host a SOA Systinet component.

14 Return to the bus' details page. The **Configuration** tab is open by default.

15 Under **Destination resources** click **Destinations**. A table of destinations opens.

16 Add the following destinations by clicking **New** in the table of destinations for each one:

Destination type	Identifier	Bus member
Queue	RFReportingExecutions	Reporting Service host
Queue	scheduleTimerQueue	SOA Systinet Platform host
Queue	taskProcessorQueue	SOA Systinet Platform host
Topic space	taskStopperTopic	SOA Systinet Platform host
Queue	PMValidationQueue	Policy Manager host

Setting Up JMS for the Reporting Service

The reporting service requires JMS messaging resources that you must set up in the WebSphere Administration Console.

To set up JMS for the reporting service:

1 Create a JMS Queue Connection Factory for sending reports.

   a Navigate to **Resources**->**JMS**->**Queue Connection Factories**. The **Queue Connection factories** page opens.

   b In the scope drop-down field, select your cell.

c Click **New**. The **Create new queue connection factory** wizard opens.

d Select **Default messaging provider**.

e Give the queue connection factory a unique, arbitrary descriptive name, such as `RF Connection Factory (Send)`.

f Give the queue connection factory the JNDI name `jms/ReportingSenderConnectionFactory`. The JNDI name must be exact!

g In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

h Click **OK** and save your changes.

2 Create a JMS Queue Connection Factory for receiving reports. Perform the same procedure you did when you created a connection factory for sending reports in Step 1, except give the receiving connection factory the JNDI name `jms/ReportingReceiverConnectionFactory`. This name must be exact.

3 Create a JMS Queue.

a Navigate to **Resources**->**JMS**->**Queues**. The **Queues** page opens.

b Click **New**. The **Create new queue** wizard opens.

c Select **Default messaging provider**.

d Give the queue a unique, arbitrary descriptive name, such as `RF Executions Queue`.

e Give the queue the JNDI name `queue/ReportingExecutions`. This name must be exact.

f In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

g In the **Queue name** drop-down field, select **RFReportingExecutions**.

h Click **OK** and save changes.

4 Create a JMS Activation for the reporting service.

a Navigate to **Resources**->**JMS**->**Activation specifications**. The **Activation specifications** page opens.

b In the scope drop-down field, select your cell.

c Click **New**. The **Activation specifications** wizard opens.

d Select **Default messaging provider**.

e Give the activation a unique, arbitrary descriptive name, such as `RF Activation`.

f Give the activation the JNDI name `jms/RFActivation`. This name must be exact.

g In the **Destination type** drop-down field, select **Queue**.

h In the **Destination JNDI name** field, type `queue/ReportingExecutions`.

i In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

j Click **OK** and save changes.

5 Create a Work Manager for the reporting service.

a Navigate to **Resources**->**Asynchronous beans**->**Work managers**. The **Work managers** page opens.

b In the scope drop-down field, select your cell.

c Click **New**. The **Create new work manager** wizard opens.

d Give the work manager a unique, arbitrary descriptive name, such as `RFWorkManager`.

e Give the work manager the JNDI name `wm/reporting`. This name must be exact.

f Click **OK** and save changes.

Setting Up JMS for Platform

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

To set up JMS for Platform:

1. Create a JMS Connection Factory for Platform.
   a. Navigate to **Resources**->**JMS**->**Connection Factories**. The **Connection factories** page opens.
   b. In the scope drop-down field, select your cell.
   c. Click **New**. The **Create new connection factory** wizard opens.
   d. Select **Default messaging provider**.
   e. Give the connection factory a unique, arbitrary descriptive name, such as `Platform Connection Factory`.
   f. Give the connection factory the JNDI name `/ConnectionFactory`. The JNDI name must be exact!
   g. In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.
   h. Click **OK** and save your changes.
   i. Click on **Connection pool properties** under **Additional properties** and set **Maximum connections** field to 100.
   j. Click **OK** again and save your changes.

2. Create the `scheduleTimerQueue` JMS Queue.
   a. Navigate to **Resources**->**JMS**->**Queues**. The **Queues** page opens.
   b. Click **New**. The **Create new queue** wizard opens.
   c. Select **Default messaging provider**.

d Give the queue a unique, arbitrary descriptive name, such as `Platform ScheduleTimerQueue`.

e Give the queue the JNDI name `queue/scheduleTimerQueue`. This name must be exact.

f In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

g In the **Queue name** drop-down field, select **scheduleTimerQueue**.

h Click **OK** and save changes.

3 Create the `taskProcessorQueue` JMS Queue. Perform the same procedure you did when you created the `scheduleTimerQueue` in Step 2, but give the `taskProcessorQueue` the JNDI name `queue/taskProcessorQueue`. This name must be exact.

4 Create the `taskStopperTopic` JMS topic.

a Navigate to **Resources**->**JMS**->**Topics**. The **Topics** page opens.

b Click **New**. The **Create new topic** wizard opens.

c Select **Default messaging provider**.

d Give the queue a unique, arbitrary descriptive name, such as `Platform TaskStopperTopic`.

e Give the queue the JNDI name `topic/taskStopperTopic`. This name must be exact.

f In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

g In the **Topic space name** drop-down field, select **taskStopperTopic**.

h Click **OK** and save changes.

5 Create a Work Manager for Platform.

a Navigate to **Resources**->**Asynchronous beans**->**Work managers**. The **Work managers** page opens.

b In the scope drop-down field, select your cell.

c Click **New**. The **Create new work manager** wizard opens.

d Give the work manager a unique, arbitrary descriptive name, such as `Platform WorkManager`.

e Give the work manager the JNDI name `/wm/platform`. This name must be exact. It does start with a / although other work manager JNDI names do not.

f Under **Thread pool properties**, set **Maximum number of threads** to 100.

g Click **OK** and save changes.

Setting Up JMS for Policy Manager

Policy Manager requires JMS messaging resources that you must set up in the WebSphere Administration Console.

To set up JMS for Policy Manager:

1 Create a JMS Queue Connection Factory for Policy Manager.

a Navigate to **Resources**->**JMS**->**Queue Connection Factories**. The **Queue connection factories** page opens.

b In the scope drop-down field, select your cell.

c Click **New**. The **Create new queue connection factory** wizard opens.

d Select **Default messaging provider**.

e Give the queue connection factory a unique, arbitrary descriptive name, such as `PM Connection Factory`.

f Give the queue connection factory the JNDI name `jms/PMConnectionFactory`. The JNDI name must be exact!

g In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

h Click **OK** and save your changes.

2 Create a JMS Queue.

a Navigate to **Resources**->**JMS**->**Queues**. The **Queues** page opens.

b Click **New**. The **Create new queue** wizard opens.

c Select **Default messaging provider**.

d Give the queue a unique, arbitrary descriptive name, such as `PM Validation Queue`.

e Give the queue the JNDI name `queue/Validation`. This name must be exact.

f In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

g In the **Queue name** drop-down field, select **PMValidationQueue**.

h Click **OK** and save changes.

3 Create a JMS Activation for Policy Manager.

a Navigate to **Resources**->**JMS**->**Activation specifications**. The **Activation specifications** page opens.

b In the scope drop-down field, select your cell.

c Click **New**. The **Activation specifications** wizard opens.

d Select **Default messaging provider**.

e Give the activation a unique, arbitrary descriptive name, such as `PM Activation`.

f Give the activation the JNDI name `jms/PMActivation`. This name must be exact.

g In the **Destination type** drop-down field, select **Queue**.

h In the **Destination JNDI name** field, type `queue/Validation`.

i. In the **Bus name** drop-down field, select the bus you created in Creating a Messaging Bus on page 91.

j. Click **OK** and save changes.

Deploying the Remaining EAR Files

After you set up JMS for all SOA Systinet components, stop and restart all WebSphere servers in the SOA Systinet host domain, including the Deployment Manager. Then deploy the SOA Systinet EAR files you created previously, following the same procedure you used to deploy the SSO service (See Deploying the SSO Service and Creating EAR Files on page 87), with the following differences:

- The EAR file names and locations are as follows:
  - REPORTING_HOME/deploy/hp-soa-systinet-reporting.ear
  - PLATFORM_HOME/deploy/hp-soa-systinet-platform.ear
  - POLICYMGR_HOME/deploy/hp-soa-systinet-policymgr.ear

- When you are deploying Platform and select deployment options as in Step 8 of Deploying the SSO Service and Creating EAR Files on page 87, also select **Allow dispatching includes to remote resources** and **Allow servicing includes from remote resources**.

- **Clustered components only:** If you are deploying Policy Manager, Platform and/or Reporting Service (but not SSO) to a cluster, you must set a cookie path for the deployed application's web module. Set the cookie path *after* you deploy the component but *before* you start the deployed component.

  To set a cookie path for a clustered application:

  1. In the Administration Console, go to **Applications**->**Enterprise Applications**. A list of deployed applications appears.
  2. Click the name of a clustered SOA Systinet application. Its details page opens.
  3. Find the **Web Module Properties** section and click **Session Management**. The **Session Management** page opens.

4 Click **Enable Cookies**. The **Cookies** page opens.

5 Type in the cookie path specific to the application:

HP SOA Systinet Platform	/soa
HP SOA Systinet Reporting	/reporting
HP SOA Systinet Policy Manager	/policymgr

6 Leave all other settings at default and click OK. You return to the **Session Management** page.

7 Find the **General Properties** section and select **Override session management**.

8 Leave all other settings at default, click **OK** and save your changes.

9 Restart all managed servers. The SOA Systinet deployments should start automatically.

10 Verify that the SOA Systinet deployments are running. Try to view their deployment descriptors in a browser window, at these addresses:

- *hostname:port*/sso/rest/descriptor
- *hostname:port*/reporting/
- *hostname:port*/soa/systinet/platform/rest/
- *hostname:port*/policymgr/rest/service/system/product-information/

10 Deploying SOA Systinet to Oracle Application Server

To deploy SOA Systinet to OAS, set up messaging resources on the server, set up SSL, create SOA Systinet EAR files, and deploy the EAR files. Deploy the SSO service EAR file and have it running before you create the EAR files for other components.

Important: If you are installing Policy Manager, use hostnames instead of IP addresses when installing OAS and deploying SOA Systinet components.

The procedures for deploying SOA Systinet to OAS are described in the following sections:

- Update JDBC Driver on page 102
- Setting Up SSL on OAS on page 102
- Allowing JNDI Lookups on page 105
- Setting log4j Properties on page 106
- Creating JDBC Resources on page 107
- Allocating Memory and Setting Java Options on page 110
- Creating JMS Resources on page 110
- Creating and Deploying SOA Systinet EARs on page 111

Update JDBC Driver

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

To update the JDBC driver for OAS:

1. Make certain OC4J is not running.
2. Open a command console in `ORACLE_HOME/product/10.1.3.1/oas1/jdbc/lib`. Rename the existing `ojdbc14dms.jar` file to `ojdbc14dms.jar.backup`.
3. In a web browser, open the Oracle JDBC driver download page [http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html] and download version 10.0.2.0 or later of `ojdbc14dms.jar`.
4. Copy the newly downloaded `ojdbc14dms.jar` to `ORACLE_HOME/product/10.1.3.1/oas1/jdbc/lib`.

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.

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

1. Launch the Oracle Wallet Manager.
   - On Windows, launch Wallet Manager from the GUI menu ***your_server_name*** +**Integrated Management tools**->**Wallet Manager**.
   - On UNIX systems, launch Wallet Manager with the `own` script in `ORACLE_HOME/bin`.
2. In the Wallet Manager GUI, click **Menu**->**Wallet**->**New**. The **Create New Wallet** wizard opens.
3. If you are asked to create a default wallet directory, select **No**.
4. Select wallet type **Standard**.

5 Type a password for the wallet. This password must be *at least* 8 characters long and contain both letters and numbers.

6 Click **Ok**. An empty wallet is created.

7 When you are prompted to create a certificate for the wallet, click **Yes**. A detail page for the certificate opens.

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

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

9 Enter other details according to your company's policy on creating certificates.

10 Click **Menu**->**Wallet**->**Save**. You are prompted for the location to save the wallet. Type `ORACLE_HOME\Apache\Apache\conf\ssl.wlt\new`. Confirm that you want to create this directory.

11 Click **Menu**->**Wallet**->**Auto Login**.

12 Click **Menu**->**Wallet**->**Save**.

13 Export the certificate request.

a Click **Menu**->**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.com.
- A corporate certification authority. Check your company's IT security guidelines.

- Yourself, with the OpenSSL tool.

16 When you obtain a signed copy of your certificate from the CA, also obtain a copy of the CA's own certificate.

17 Reopen Wallet Manager and the wallet you created.

18 Click **Menu**->**Operations**->**Import Trusted Certificate**. Import the certificate of the CA that signed your certificate.

19 Click **Menu**->**Operations**->**Import User Certificate**. Import the signed copy of your certificate.

20 Click **Menu**->**Wallet**->**Save** then exit Wallet Manager.

21 Set up Oracle HTTP server to use the wallet you created.

a Open `ORACLE_HOME\Apache\Apache\conf\ssl.conf` in an editor.

b Change the `SSLWallet` directive to point to the newly generated wallet file at `ORACLE_HOME\Apache\Apache\conf\ssl.wlt\new`.

To enable SSL communication between SOA Systinet applications, the Oracle Application Servers that host the SOA Systinet applications must trust the certificate of the CA that signed your Oracle HTTP server SSL user certificate. You obtained this CA certificate in Step 16. There are two approaches to having OAS trust this certificate:

- Have every new OC4J instance trust the CA certificate. For every OC4J instance to trust this certificate, import it into the trust store of the JDK used by OC4J. By default, this JDK is bundled with Oracle Application Server. To import the CA certificate:

  1 Run `ORACLE_HOME/jdk/bin/keytool` with the following parameters: `-import -alias <ca_certificate_alias> -file <ca_certificate_file> -keystore ORACLE_HOME\jdk\jre\lib\security\cacerts -noprompt -trustcacerts`

  2 Restart OC4J. Run `ORACLE_HOME/opmn/bin/opmnctl shutdown` then `ORACLE_HOME/opmn/bin/opmnctl startall`.

- Configure a particular OC4J instance to use a custom truststore with the CA certificate. To configure this instance:

1. Run the `ORACLE_HOME/jdk/bin/keytool` command with the following parameters: `-import -alias <ca_certificate_alias> -file <ca_certificate_file> -keystore <keystore_other_than_jdk_keystore (can be new keystore)> -noprompt -trustcacerts` If you create a new keystore, note its password.

2. Open `ORACLE_HOME/opmn/opmn.xml` in an editor.

3. For a defined OC4J process-type, change `java-options` in the `start-parameters` element to include -Djavax.net.ssl.trustStore=<*path_to_generated_trust_store*> and -Djavax.net.ssl.trustStorePassword=<*keystore_password*>

4. Save `ORACLE_HOME/opmn/opmn.xml` and restart OC4J.

Allowing JNDI Lookups

SOA Systinet Platform creates threads that use JNDI lookups. By default, OC4J does not allow JNDI lookups in application-created threads. Add an option to the command that starts OC4J instances that host SOA Systinet Platform so that OC4J will allow JNDI lookups. You can do this in the Oracle administration console or by editing `opmn.xml`

To enable JNDI lookups in the Oracle Administration Console:

1. Start the Adminstration Console for the OC4J instance that hosts Platform.

2. Open the **Administration** tab.

3. Navigate to **Administration Tasks**->**Properties**->**Server Properties**->**Command line options ...**

4. Add the Java option `-Doc4j.userThreads=true`.

To enable JNDI lookups by editing opmn.xml:

1. Open `ORACLE_HOME/opmn/conf/opmn.xml` in an editor.

2. Add the Java option `-Doc4j.userThreads=true` to the start parameters of process defintions of 0C4J instances that will host SOA Systinet Platform. A snippet of such a process definition follows:

```
<process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
```

```
        <category id="start-parameters">
            <data id="java-options" value="-Xrs -server -XX:MaxPermSize=128M -ms512M -mx1300M
                    -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>
```

3 Save your changes.

Setting log4j Properties

SOA Systinet uses log4j.logging. SOA Systinet provides a copy of `log4j.jar` and puts it on the classpath, but you must create your own log4j configuration file. Save the log4j configuration file as `log4j.properties` in ORACLE_HOME/j2ee/home/lib/ext.

Example 1 on page 107 is a log4j configuration you can copy and paste. This configuration instructs log4j to log to both console and a file named `log4j.log`, which log4j creates by default in DOMAIN_HOME. This configuration also strips overly verbose log4j categories down to ERROR and FATAL logs.

Example 1: Log4j Configuration File

```
# console appender
log4j.appender.stderr=org.apache.log4j.ConsoleAppender
log4j.appender.stderr.Target=System.err
log4j.appender.stderr.layout=org.apache.log4j.PatternLayout
log4j.appender.stderr.layout.ConversionPattern=%p: %c{2} - %m%n

# file appender
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.maxFileSize=100KB
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

# save all logs in stderr and file
log4j.rootLogger=INFO,stderr,file

# limit categories that are too verbose
log4j.category.org.apache.xml.security=ERROR,stderr
log4j.additivity.org.apache.xml.security=true
log4j.category.org.apache.beehive=ERROR,stderr,file
log4j.additivity.org.apache.beehive=true

#configure hibernate (prevents certain spurious error messages)
log4j.category.org.hibernate=OFF,stderr,file
log4j.additivity.org.hibernate=true
```

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.

To prepare to create JDBC resources:

1. Start OC4J. Run `ORACLE_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.

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

To create a connection pool:

1 Navigate to **Administration Tasks->Services->JDBC Resources**.

2 Find the **Connection Pools** section and click **Create**. A wizard opens.

3 Leave **Application** as **default**, select **New connection pool** and 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 fill in the following connection property fields:

Field name	**Value**
Driver Type	Thin
DB Host Name	*Type database host name.*
DB Listener Port	1521 *(Oracle default)*
DB Identifier Type	SID
SID/Service Name	*Type sid.*
Username (under Credentials section)	*Type database username. Enter the same database username when you run SOA Systinet installers.*
Use Cleartext Password / Password	*Type database password. Enter the same database password when you run SOA Systinet installers.*

7 Click **Test Connection**. This tests the connection properties you entered in Step 6. Double-check the connection properties if the test fails.

8 Open the **Attributes** tab and fill in the following fields:

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, use it to create a datasource.

To create a datasource:

1 Navigate to **Administration Tasks**->**Services**->**JDBC Resources**.

2 Find the **Data Sources** section and click **Create**. A wizard opens.

3 Leave **Application** as **default**, select **Managed data source** and click **Continue**. A details page for a new datasource opens.

4 Fill in the following fields:

Field name	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 *(default)*

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

5 Click **Finish**.

Allocating Memory and Setting Java Options

Increase the memory Maximum Permanent size to 512 MB. If the server will host Reporting or Policy Manager, also set `java.awt.headless` to "true."

To increase Max Perm Size:

1 Navigate to **Administration Tasks**->**Properties**->**Server Properties**.

2 In the **Options** table, set `-XX:MaxPermSize=512M`.

3 If the server will host Reporting or Policy Manager, set `java.awt.headless=true`.

4 Click **Apply**.

5 Restart OC4J. Run `ORACLE_HOME/opmn/bin/opmnctl shutdown` then `ORACLE_HOME/opmn/bin/opmnctl startall`.

Creating JMS Resources

Each SOA Systinet application requires connection factories and destinations.

To create JMS connection factories for SOA Systinet:

1 Navigate to **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 Step 2 and this step until you have created all the connection factories.

Connection Factory Type	JNDI Location	Check "XA Enabled"?
Queue	jms/ReportingSenderConnectionFactory	Yes
Queue	jms/ReportingReceiverConnectionFactory	Yes

Connection Factory Type	JNDI Location	Check "XA Enabled"?
Unified	/ConnectionFactory	No
Queue	jms/PMConnectionFactory	Yes

To create JMS destinations for SOA Systinet:

1 Navigate to **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 Step 2 and this step 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

Creating and Deploying SOA Systinet EARs

After configuring JDBC and JMS resources, use the SOA Systinet installers to create EAR files and deploy them with the Oracle Application Server Control. Create, deploy and run the SSO service EAR before creating other EAR files.

Important: If you are *re-deploying* a SOA Systinet application to OAS, *undeploy* the existing application and restart OC4J before deploying the updated EAR file.

To deploy SOA Systinet:

1. Create the SSO service EAR file as described in Installing the Single Sign-On Service on page 117.
2. Start OC4J. Run `ORACLE_HOME/opmn/bin/opmnctl startall`
3. Open the Application Server Control (`http://localhost/em`).
4. Log in as the OC4J administrator.
5. Click the link to your OC4J instance (by default named **Home**) in the tree view on the main page.
6. Open the **Applications** tab.
7. Click **Deploy**. The deployment wizard opens.
8. Enter the path to the file `SSO_HOME/deploy/hp-soa-systinet-sso.ear`. You can browse for this path by clicking **Location on Server**.
9. Keep the option **Automatically create a new deployment plan**.
10. Click **Next**. The next page opens.
11. Type in the application name `HP SOA Systinet SSO`.
12. Leave all other fields at default and click **Next**. A page with the deployment details opens.
13. Review the deployment details and if they are correct click **Deploy**. SSO service is deployed.
14. Check that the deployment succeeded (your URLs may differ):
    - View `http://localhost:80/sso/rest/descriptor` in a browser. An XML file must be returned.
    - View `http://localhost:80/sso/account/wsdl` in a browser. A WSDL file must be returned.

- Run `SSO_HOME/bin/ssoconfig get -d partners -s http://localhost:80/sso -u admin -p changeit`. No exceptions should be thrown. Note that 'admin' and 'changeit' are the default admin name and password for SSO service.

15 Run the other SOA Systinet installers as described in Part IV, “Installing SOA Systinet Components”. This creates the following EAR files:

- REPORTING_HOME/deploy/hp-soa-systinet-reporting.ear
- PLATFORM_HOME/deploy/hp-soa-systinet-platform.ear
- POLICYMGR_HOME/deploy/hp-soa-systinet-policymgr.ear

16 Deploy all undeployed EAR files, following the same procedure you used to deploy the SSO service EAR. The location and names of each EAR file differ for each component as listed in Step 15.

17 Check that the deployment succeeded (your URLs may differ):

- View `http://localhost:80/soa/systinet/platform/rest/repository/?acl` in a browser.
- View `http://localhost:80/soa/systinet/platform/restBasic/repository/?acl`. Input admin credentials and check that effective permissions are read and write
- Open the SOA Systinet console at `http://localhost:80/soa/systinet/platform/web`. No exceptions should be thrown. Sign in and sign out. Create and run a task to test JMS connections.

Part IV. Installing SOA Systinet Components

This part describes how to use the SOA Systinet installation wizards. For the JBoss application server, the SOA Systinet installation wizards handle deployment automatically, and the server environment is set up when you start JBoss with the SOA Systinet `serverstart` script. For other application servers, the SOA Systinet installation wizards create EAR files, which are then installed with that application server's deployment tools.

- Installing the Single Sign-On Service on page 117
- Installing Reporting Service on page 129
- Installing Platform on page 135
- Installing Policy Manager on page 143
- Using Silent Installation on page 149

11 Installing the Single Sign-On Service

This chapter contains the following sections describing the procedures of installing the Single Sign-On (SSO) Service:

Running the SSO Installer on page 117. Executing the installer .jar file and the top-level procedure for installing SSO.

LDAP Accounts Integration on page 121. An explanation of the information you need to provide when you use LDAP backend accounts with the SSO service, including mapping between SOA Systinet and LDAP properties.

In the installation process, the SSO service must be running when you install the other components.

Running the SSO Installer

To install the SOA Systinet Single Sign-On (SSO) service:

1. Install the target J2EE application server. Servers other than JBoss require setup before you can install SSO.
2. Make sure the J2EE server is not running.
3. Execute the file `hp-soa-systinet-sso-2.52.jar`, located on the installation CD or in your distribution directory. On all operating systems, the command **java -jar hp-soa-systinet-sso-2.52.jar** executes the file. Executing this file opens the installation wizard.

Note: On Windows networks, copy the installer jar to your local file system and run it from there instead of from the network.

To see the command-line options, run **java -jar hp-soa-systinet-sso-2.52.jar --help**. Use these options to set up and run a silent installation (see Using Silent Installation on page 149).

Important: If you are using SOA Systinet with a manually created database, you must run a silent installation.

4 The **Welcome** screen opens with hardware and software requirements. Read this carefully before clicking **Next**, which opens the license page.

5 Read and accept the license. Click **Next** to proceed to the **Installation Folder** page.

6 Type or browse the location you want for your SOA Systinet SSO server installation folder. This location is referred to as SSO_HOME. The default location is `C:\Program Files\Hewlett-Packard\Systinet\SSO`. Each SOA Systinet component requires a separate installation subfolder. Click **Next** and the installer unpacks the distribution files to the chosen location.

7 Select the type of J2EE application server to which you are installing the SSO service. Not all installation steps are performed for every application server. The subsequent installation path differs as shown in Table 8 on page 118.

Table 8. Single Sign-on Installation Path by Application Server

Application Server	Description	Steps Performed in Default Installation
JBoss	By default, complete, automated deployment to the server is performed.	All (See Step 8.)
WebLogic, OAS or WebSphere	A deployable EAR file is created. This must be deployed with application server tools.	Step 10, Step 11, Step 12, Step 13, Step 18

8 Select either **Advanced** or **Default** deployment. Default deployment performs all steps that can be performed for the chosen application server. For advanced deployment, select a subset of steps to perform:

- Password Encryption, Step 9
- Database Setup, Step 10

- Configuration Table Setup, Step 11
- Endpoint Properties, Step 12
- SSO Setup, Step 13
- Application Server Properties (JBoss only), see Step 14
- Datasource Setup (JBoss only), Step 15
- Deployment (JBoss only), Step 16
- SSL Setup (JBoss only), Step 17

9 Select whether to encrypt passwords. Password encryption is recommended for production environments. If you set a passphrase for encrypting your passwords, include `-Dpassword.encryption.passphrase=`*passphrase* on the command line for non-interactive installation (Using Silent Installation on page 149) or the command-line tools that require authentication. To change the passphrase after installation, run the Setup tool in the **Advanced** scenario and select the **Password Encryption** option.

Caution: If you install more than one SOA Systinet component onto the same application server, so that they share the same db tablespace, you have the following passphrase limitations:

- You must use the same passphrase for all components sharing a tablespace.
- You cannot change the shared passphrase with the Setup tool. If you need to change a passphrase shared by multiple SOA Systinet components, contact HP professional services.

Note: Do not use the administrator password as the passphrase.

If you later lose the passphrase, contact HP professional services for assistance.

10 Set up the database.

a. Select **Create Database** or **Create Schema**, which creates a schema in an existing database. Click **Next** to proceed. Consult your database administrator for instructions. See Database Installation Types on page 27 for details.

b. Select which type of database you are using, such as Oracle or DB2.

c. Type in database parameters. Please see Table 7 on page 29 for details. All SOA Systinet components must use the same database.

> **Note:** If you are using the Oracle database, you can enter the connection string instead of the server address, port and database name. You still must enter the user credentials.

d. Type in or browse to the full paths of the JDBC driver JAR/ZIP file to be installed to SSO, separated by commas. For Oracle, this should be `ojdbc14.jar`. For DB2, these should be the two files `db2jcc.jar` and `db2jcc_license_cu.jar`. If you do not have these files, contact your database administrator.

e. The installer copies the JDBC drivers and verifies the connection to the database.

11. The installer verifies that a configuration table is available.

12. Specify the endpoint properties: Hostname, HTTP and HTTPS port numbers, whether or not to use HTTPS transport, and the web context. The default values are given for JBoss. If you are installing on another application server, type the port numbers of the managed server or cluster that will host the SSO service. If communication goes through a proxy server, use the proxy server's hostname and port numbers .

> **On JBoss:** If you change the port numbers from their default values, you must also change the application server configuration to use these ports. See Configuring JBoss When SOA Systinet Uses Non-default Ports on page 56.

13. Set up the SSO parameters that will be used by other SOA Systinet servers to communicate with the SSO server.

a Specify SOA Systinet administrator username and password. The default password is `changeit`.

b Select whether to store user accounts on the **Database** or on an **LDAP** backend. If you store user accounts on an LDAP backend, enter the LDAP service properties. The relationship between SOA Systinet and LDAP properties is described in LDAP Accounts Integration on page 121. Please also consult your LDAP administrator.

   For installation to the WebLogic, Oracle and WebSphere application servers, this is the final step before the EAR file is created in Step 18.

14 (JBoss only) Type or browse to the **JBoss Installation Folder** and type the **JBoss Configuration**. If installing SSO standalone, the configuration should be `default` or point to a copy you made of `JBOSS_HOME/server/default`. If installing SSO to a cluster, the configuration should be `all` or point to a copy you made of `JBOSS_HOME/server/all`.

15 (JBoss only) The installer verifies the datasource.

16 (JBoss only) The installer verifies the application server connection.

17 (JBoss only) The installer verifies that the necessary client truststore is present for SSL communication.

18 Depending on the type of installation, the installer either creates a deployable EAR file in `SSO_HOME/deploy` or deploys SSO to the JBoss application server.

LDAP Accounts Integration

When you install the Single Sign-On (SSO) service (see Installing the Single Sign-On Service on page 117), you can select to store user accounts on an external LDAP server. This chapter describes how to integrate accounts from an LDAP server into SOA Systinet. It includes the following sections:

- Automatic Service Discovery on page 122. A brief explanation of automatic service discovery and its implications
- LDAP Service Properties on page 123. A list of JNDI properties of the LDAP server that must be known to the SSO service.

- LDAP with a Single Search Base on page 123. One of two use scenarios (the other being LDAP with multiple search bases). The single search base scenario is very simple. There is only one LDAP server. All identities are stored under a single search base.
- LDAP with Multiple Search Bases on page 126. One of two use scenarios (the other being LDAP with a single search base). In the multiple search bases scenario, there is also only one LDAP server, but it has multiple search bases mapped to a domain. The domain is a specified part of the user's login name (that is, `DOMAIN/USERNAME`). All users must specify the domain name in the login dialog box. When you manage accounts or groups, use the `DOMAIN/USERNAME` format. If no domain is set, searches are performed across all domains.
- LDAP over SSL/TLS on page 126. Various scenarios for enabling communication over SSL between the SSO service and the LDAP server.

Important: TAdministrator account must not be stored in the LDAP. We strongly recommend that users stored in `account_list.xml` (by default, only administrator) should not be in LDAP. If you really need to have users from LDAP in the file `SSO_HOME/conf/system/account_list.xml`, delete password items from the file and change of all the accounts' properties according to LDAP. The `account_list.xml` file contains a list of users that can be logged into SOA Systinet without connection to the database.

Sometimes SOA Systinet displays various warnings into logs. We recommend suppressing account/group LDAP integration warnings. To suppress these warnings, open the files `SSO_HOME/conf/system/directory.xml` and `SSO_HOME/conf/system/group_core.xml` and set all instances of the attribute `suppressWarnings` to `true`.

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 will be 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 be sure of the Java version you are using.

LDAP Service Properties

To integrate external accounts, during Single Sign-On (SSO) service installation select **LDAP** in the account provider panel.

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/1.4.2/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/1.4.-2/docs/api/javax/naming/Context.-html#PROVIDER_URL
Initial Naming Factory	Java class for the initial naming factory	http://java.sun.com/j2se/1.4.-2/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/1.4.-2/docs/api/javax/naming/Context.-html#SECURITY_PRINCIPAL
Password	Password of security principal	http://java.sun.com/j2se/1.4.-2/docs/api/javax/naming/Context.-html#SECURITY_CREDENTIALS
Security Protocol	Name of the security protocol. Default is "simple."	http://java.sun.com/j2se/1.4.-2/docs/api/javax/naming/Context.-html#SECURITY_PROTOCOL

LDAP with a Single Search Base

The installation consists of the following steps:

1 Specify user/account search properties.

2 Map SOA Systinet user search properties to LDAP properties.

3 Specify group search properties.

4 Map SOA Systinet group search properties to LDAP properties.

Users and groups have the same properties. These properties are described in Table 9 on page 124

Table 9. SOA Systinet User and Group Search 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.	
Search Base	LDAP will be searched from this base including the current LDAP node and all possible child nodes.	
Search Scope	Object Scope	Only the search base node will be searched.
	One-level Scope	Only direct sub-nodes of the search base (entries one level below the search base) will be searched. The base entry is not included in the scope.
	Subtree Scope	The search base and all its sub-nodes will be searched
Results Limit	Number of items returned when searching LDAP. If more than this number of results are returned by an LDAP search an error occurs.	

The following user account properties can be mapped from an LDAP server:

```
java.lang.String loginName
java.lang.String email
java.lang.String fullName
java.lang.String languageCode
java.lang.String password
java.lang.String description
java.lang.String businessName
java.lang.String phone
java.lang.String alternatePhone
java.lang.String address
java.lang.String city
java.lang.String stateProvince
java.lang.String country
java.lang.String zip
java.util.Date expiration
java.lang.Boolean expires
java.lang.Boolean external
```

```
java.lang.Boolean blocked
java.lang.Integer businessesLimit
java.lang.Integer servicesLimit
java.lang.Integer bindingsLimit
java.lang.Integer tModelsLimit
java.lang.Integer assertionsLimit
      java.lang.Integer subscriptionsLimit
```

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

```
java.lang.String name
      java.lang.String owner
      java.lang.String description
      java.lang.Boolean privateGroup
      java.lang.String member
```

Important: The platform account property **dn** specifies the LDAP distinguished name. The value depends on the LDAP vendor.

- On the Sun ONE Directory Server, the value is **entryDN**
- On Microsoft Active Directory, the value is **distinguishedName**

User account properties that you specify when mapping to LDAP are treated as read-only in SOA Systinet.

If an optional property (such as email) does not exist in LDAP, then the property value is set according to the default account or group. The default account is specified in the config file `SSO_HOME/conf/system/account_core.xml`. The default group (groupInfo) is specified in the config file whose name is `group_core.xml`.

You can specify mapping between SOA Systinet group properties and LDAP properties. You can add rows by clicking **Add**. To edit an entry, double click on the value you wish to edit.

If a property (such as description) does not exist in LDAP then property value is set according to the default group.

LDAP with Multiple Search Bases

The installation consists of the following steps:

1. Specify the domain delimiter, domain prefix and postfix. These properties are used to dynamically specify domains.
2. Enable/Disable domains. In this step you can statically specify additional domains or disable domains.
3. Specify and map user/account search properties and group search properties as with single search bases. See LDAP with a Single Search Base on page 123.

Domain properties are described in Table 10 on page 126.

Table 10. SOA Systinet Domain Properties

Property	Description
Domain Delimiter	Specifies the character that delimits domain and user name.
Domain Prefix, Domain Postfix	Allows the dynamic specification of domains. Domains are searched using the following pattern: `{domain prefix}domain_name{domain postfix}{search base}` where **{}** curly brackets indicate the value of the property whose name is contained in the brackets.

LDAP over SSL/TLS

It is only a matter of configuration to set up LDAP over *SSL* (or*TLS*) with a directory server of your choice. We 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`.

Be 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 will get an exception during startup of SOA Systinet. It will inform 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 LDAPS described in LDAP over SSL/TLS on page 126 inherits the default rule for establishing trust from JSSE (the Java implementation of SSL/TLS). This is based on trust stores.

The trust store for SOA Systinet is located in `SSO_HOME/conf/client.truststore` and the certificate for the LDAP server or its certification authority should be added to it.

To add the LDAP certificate to the SOA Systinet trust store, contact the administrator of the LDAP server and get the certificate of the server or the certificate of the authority that signed it, then Import the certificate into the SSO service trust store using the Java keytool:

keytool -import -trustcacerts -alias *`alias`* -file *`file`* -keystore *`keystore`* -storepass *`storepass`*

The parameters in the `keytool` command are as follows:

Parameter	Description
`alias`	A mandatory, unique alias for the certificate in the trust store;
`file`	The file containing the certificate (usually with .crt extension);
`keystore`	The SOA Systinet keystore file (`SSO_HOME/conf/client.truststore`).

Parameter	Description
`storepass`	A password designed to protect the keystore file from tampering. The password for the SOA Systinet keystore is the *SSL Certificate Password* set during installation. The default is `changeit`.

12 Installing Reporting Service

After installing the SSO service (see Installing the Single Sign-On Service on page 117), install the reporting service. The SSO service must be running when you run the Reporting Service installer. The installation process is identical to reconfiguring an installed reporting service with the Setup tool (see the Administration Guide), which starts at Step 8.

To install the reporting service:

1. Install the target J2EE application server. Except on JBoss, the reporting service requires a separate managed server or cluster from other SOA Systinet components.

2. Start the J2EE server hosting the SSO service. On JBoss, use the `SSO_HOME/bin/serverstart` script.

3. Execute the file `hp-soa-systinet-reporting-standard-2.52.jar` , located on the installation CD or in your distribution directory. In Windows, you can double-click on it in an exploration window. On all operating systems, it launches with the command **java -jar hp-soa-systinet-reporting-standard-2.52.jar** . Executing this file opens the installation wizard.

   Note: On Windows networks, copy the installer jar to your local file system and run it from there instead of from the network.

   To see the command-line options, run **java -jar hp-soa-systinet-reporting-standard-2.52.jar --help** . Use these options to set up and run a silent installation (see Using Silent Installation on page 149).

   Important: If you are using SOA Systinet with a manually created database, you must run a silent installation.

4. The welcome screen opens with hardware and software requirements. Read this carefully before clicking **Next**, which opens the license page.

5 Read and accept the license. Click **Next** to proceed to the **Installation Folder** page.

6 Type or browse the location you want for your SOA Systinet reporting server installation folder. This location is referred to as REPORTING_HOME. The default location is `C:\Program Files\Hewlett-Packard\Systinet\Reporting`. Each SOA Systinet component requires a separate installation subfolder. Click **Next** and the installer unpacks the distribution files to the chosen location.

7 Select the type of J2EE application server to which you are installing the reporting service. Not all installation steps are performed for every application server. The subsequent installation path differs as shown in Table 11 on page 130.

Table 11. Reporting Service Installation Path by Application Server

Application Server	Description	Steps Performed in Default Installation
JBoss	By default, complete, automated deployment to the server is performed.	All (See Step 8.)
WebLogic, Oracle AS or WebSphere	A deployable EAR file is created. This must be deployed with application server tools.	Step 10, Step 11, Step 12, Step 13, Step 14Step 21

8 Select either advanced or default deployment. Default deployment performs all steps that can be performed for the chosen application server. For advanced deployment, select a subset of steps to perform:

- Password Encryption, Step 9
- Database Setup, Step 10
- Configuration Table Setup, Step 11
- Endpoint Properties, Step 12
- SSO Identity Setup, Step 13
- Application of Reporting Service Extensions, Step 14

- Reporting Service Extensions Data Import, Step 15
- Application Server Properties (JBoss only), Step 16
- Datasource Setup (JBoss only), Step 17
- JMS Setup (JBoss only), Step 18
- Deployment (JBoss only), Step 19
- SSL Setup (JBoss only), Step 20

9 Select whether to encrypt passwords. Password encryption is recommended for production environments. If you set a passphrase for encrypting your passwords, include `-Dpassword.encryption.passphrase=`*passphrase* on the command line for non-interactive installation (Using Silent Installation on page 149) or the command-line tools that require authentication. To change the passphrase after installation, run the Setup tool in the **Advanced** scenario and select the **Password Encryption** option.

Caution: If you install more than one SOA Systinet component onto the same application server, so that they share the same db tablespace, you have the following passphrase limitations:

- You must use the same passphrase for all components sharing a tablespace.
- You cannot change the shared passphrase with the Setup tool. If you need to change a passphrase shared by multiple SOA Systinet components, contact HP professional services.

Do not use the administrator password as the passphrase.

If you later lose the passphrase, contact HP professional services for assistance.

10 Set up the database.

a Choose whether to create a new database or create a new schema in an existing database. Consult your database administrator for instructions. Click **Next** to proceed. Please see Database Installation Types on page 27 for details.

b Indicate which type of database you are using, such as Oracle or DB2.

c Type in database parameters and click **Next**. Please see Table 7 on page 29 for details. All SOA Systinet components must use the same database.

> **Note:** If you are using the Oracle database, you can enter the connection string instead of the server address, port and database name. You still must enter the user credentials.

d Type in or browse to the full paths of the JDBC driver .jar/.zip file to be installed to Reporting, separated by commas. For Oracle, this should be `ojdbc14.jar`. For DB2, these should be the two files `db2jcc.jar` and `db2jcc_license_cu.jar`. If you do not have these files, contact your database administrator.

e The installer now copies the JDBC drivers and verifies the connection to the database.

11 The installer now verifies that a configuration table is available.

12 Specify the endpoint properties: Hostname, HTTP and HTTPS port numbers, whether or not to use HTTPS transport, and the web context. The default values are given for JBoss. If you are installing on another application server, type the port numbers of the managed server or cluster that will host the reporting service.

If communication goes through a proxy server, use the proxy server's hostname and port numbers (see Proxy Setup on page 165). If SOA Systinet components should not use HTTPS to communicate between each other, unselect **Use HTTPS**.

> **On JBoss:** If you change the port numbers from their default values, you must also change the application server configuration to use these ports. See Configuring JBoss When SOA Systinet Uses Non-default Ports on page 56.

13 Type in the following information:

SSO Service URL	The URL of the SSO Service. If SOA Systinet components communicate through a proxy server, use the proxy server's URL. (If a proxy server is used for external communication but SOA Systinet components communicate with each other directly, use the SSO service host's URL.)
Configuration Service Admin Name	Name and password of the SSO Service administrator. The default password is `changeit`.
Configuration Service Admin Password	
Identity Name	Arbitrary name of the identity for the reporting server you are creating on the SSO Server. Default is `reporting`.
Identity Password	The password of the SSO identity you are creating. The default password is `changeit`.

14 (Setup tool only) Applies extensions from the `/extensions` directory to the .ear file.

15 The installer now imports extension data from the extension .jar files into the database.

For installation to the WebLogic, Oracle and WebSphere application servers, this is the final step before the EAR file is created in Step 21.

16 (JBoss only) Indicate path and configuration type of JBoss server. If installing Reporting standalone, the configuration should be `default` or point to a copy you made of `JBOSS_HOME/server/default`. If installing Reporting to a cluster, the configuration should be `all` or point to a copy you made of `JBOSS_HOME/server/all`.

17 (JBoss only) The installer now verifies the datasource.

18 (JBoss only) The installer now verifies JMS setup.

19 (JBoss only) Installer now verifies application server properties.

20 (JBoss only) The installer now verifies that the necessary client truststore is present for SSL communication.

21 Depending on the type of installation, the installer either creates a deployable EAR file in `REPORTING_HOME/deploy` or deploys the reporting service to the JBoss application server.

13 Installing Platform

After installing the SSO and reporting services (see Installing the Single Sign-On Service on page 117 and Installing Reporting Service on page 129), install the SOA Systinet Platform service. The SSO service must be running when you run the Platform installer. The installation process is identical to reconfiguring an installed SOA Systinet Platform service with the Setup tool (see the Administration Guide), which starts at Step 8.

To install the SOA Systinet Platform service:

1. Install the target J2EE application server. You can use the same J2EE server that hosts the SSO and/or the reporting service.
2. Start the J2EE server hosting the SSO service. On JBoss, use the `SSO_HOME/bin/serverstart` script.
3. Execute the file `hp-soa-systinet-platform-standard-2.52.jar` , located on the installation CD or in your distribution directory. In Windows, you can double-click on it in an exploration window. On all operating systems, it launches with the command **java -jar hp-soa-systinet-platform-standard-2.52.jar.** Executing this file opens the installation wizard.

   Note: On Windows networks, copy the installer jar to your local file system and run it from there instead of from the network.

   To see the command-line options, run **java -jar hp-soa-systinet-platform-standard-2.52.jar --help.** Use these options to set up and run a silent installation (see Using Silent Installation on page 149).

   Important: If you are using SOA Systinet with a manually created database, you must run a silent installation.

4 The welcome screen opens with hardware and software requirements. Read this carefully before clicking **Next**, which opens the license page.

5 Read and accept the license. Click **Next** to proceed to the **Installation Folder** page.

6 Type or browse the path to your SOA Systinet Platform installation folder and click **Next**. The default path is `C:\Program Files\Hewlett-Packard\Systinet\Platform`. Each SOA Systinet component requires a separate installation subfolder. Click **Next** and the installer unpacks the distribution files to the chosen location.

7 Select the type of J2EE application server to which you are installing SOA Systinet Platform. Not all installation steps are performed for every application server. The subsequent installation path differs as shown in Table 12 on page 136.

Table 12. SOA Systinet Platform Installation Path by Application Server

Application Server	Description	Steps Performed in Default Installation
JBoss	By default, complete, automated deployment to the server is performed.	All (See Step 8.)
WebLogic, Oracle AS or WebSphere	A deployable EAR file is created. This must be deployed with application server tools.	Step 10, Step 11, Step 12, Step 13, Step 14, Step 15, Step 16, Step 21, Step 23, Step 24

8 Select either advanced or default deployment. Default deployment performs all steps that can be performed for the chosen application server. For advanced deployment, select a subset of steps to perform:

- Password Encryption, Step 9
- Database Setup, Step 10
- Configuration Table Setup, Step 11
- Endpoint Properties, Step 12

- SSO Identity Setup, Step 13
- Repository Import, Step 14
- UI Perspective Import, Step 15
- Reporting Server Connection, Step 16
- Application Server Properties (JBoss only), Step 17
- Datasource Setup (JBoss only), Step 18
- SSL Setup (JBoss only), Step 19
- JMS Setup (JBoss only), Step 20
- SMTP Properties, Step 21
- Deployment (JBoss only), Step 22
- Client Package Creation, Step 23

9 Select whether to encrypt passwords. Password encryption is recommended for production environments. If you set a passphrase for encrypting your passwords, include `-Dpassword.encryption.passphrase=`*`passphrase`* on the command line for non-interactive installation (Using Silent Installation on page 149) or the command-line tools that require authentication. To change the passphrase after installation, run the Setup tool in the **Advanced** scenario and select the **Password Encryption** option.

Caution: If you install more than one SOA Systinet component onto the same application server, so that they share the same db tablespace, you have the following passphrase limitations:

- You must use the same passphrase for all components sharing a tablespace.
- You cannot change the shared passphrase with the Setup tool. If you need to change a passphrase shared by multiple SOA Systinet components, contact HP professional services.

Do not use the administrator password as the passphrase.

If you later lose the passphrase, contact HP professional services for assistance.

10 Set up the database.

a Select whether to create a new database, create a new schema in an existing database or configure an existing database and schema. If you are using the Setup tool, you can also drop a database or schema. Click **Next** to proceed. Consult your database administrator for instructions. See Database Installation Types on page 27 for details.

b Select which type of database you are using, such as Oracle 10 or DB2.

c Type in database parameters. Please see Table 7 on page 29 for details. All SOA Systinet components must use the same database.

Note: If you are using the Oracle database, you can enter the connection string instead of the server address, port and database name. You still must enter the user credentials.

d Type in or browse to the full paths of the JDBC driver .jar/.zip file to be installed to Platform, separated by commas. For Oracle, this should be `ojdbc14.jar`. For DB2, these should be the two files `db2jcc.jar` and `db2jcc_license_cu.jar`. If you do not have these files, contact your database administrator.

e The installer now copies the JDBC drivers and verifies the connection to the database.

11 The installer now verifies that a configuration table is available.

12 Specify the endpoint properties:

- Hostname
- HTTP and HTTPS port numbers

- Whether or not to use HTTPS transport
- Whether or not to verify SSL certificates
- The web context

The default values are given for JBoss. If you are installing on another application server, type the port numbers of the managed server or cluster that will host Platform.

If communication goes through a proxy server, use the proxy server's hostname and port numbers (see Proxy Setup on page 165). If SOA Systinet components should not use HTTPS to communicate between each other, unselect **Use HTTPS**.

Certificate validation is enabled by default. Select to disable it **only** if you are using this installation in an evaluation or development environment, where you do not need to trust the certificates of HTTPS communication partners (such as other SOA Systinet components or HP SOA Systinet Registry). Disabling certificate verification can simplify installation in these non-critical environments. You can also disable certificate verification after installation in the UI (see the Administrator Guide.

On JBoss: If you change the port numbers from their default values, you must also change the application server configuration to use these ports. See Configuring JBoss When SOA Systinet Uses Non-default Ports on page 56.

13 Type in the following information:

SSO Service URL	The URL of the SSO Service. If SOA Systinet components communicate through a proxy server, use the proxy server's URL. (If a proxy server is used for external communication but SOA Systinet components communicate with each other directly, use the SSO service host's URL.)
Configuration Service Admin Name	Name and password of the SSO Service administrator. The default password is `changeit`.
Configuration Service Admin Password	

Identity Name	Arbitrary name of the identity for the SOA Systinet Platform server you are creating on the SSO Server. Default is `platform`.
Identity Password	The password of the SSO identity you are creating. The default password is `changeit`.

14 Select to either install the default, initial bootstrap image or to import a custom image exported from another SOA Systinet Platform server.

15 The installer now verifies UI perspective import.

16 In the **Reporting Server Connection** panel, select the SSO partner identity name of the reporting service component that you are linking to this platform component. The default partner identity name you want is **reporting**. You can also choose to use the secure SSO https URL .

If you are installing on a WebLogic, Oracle or WebSphere server, go to Step 21.

17 (JBoss only) Indicate path and configuration type of J2EE application server. If installing Platform standalone, the configuration should be `default` or point to a copy you made of `JBOSS_HOME/server/default`. If installing Platform to a cluster, the configuration should be `all` or point to a copy you made of `JBOSS_HOME/server/all`.

18 (JBoss only) The installer now verifies the datasource.

19 (JBoss only) The installer now verifies the existence of the necessary application server configuration files for SSL communication.

20 (JBoss only) The installer now verifies JMS setup.

21 If you want SOA Systinet to send notifications over email, type in SMTP server authentication details. Mail service must be configured separately on the J2EE server.

If you are installing on a WebLogic, Oracle or WebSphere server, go to Step 23.

22 (JBoss only) Installer now verifies application server properties.

23 The installer now verifies the existence of necessary client package files.

24 Depending on the type of installation, the installer either creates a deployable EAR file in `PLATFORM_HOME/deploy` or deploys SOA Systinet Platform to the JBoss application server.

14 Installing Policy Manager

After installing the SSO, reporting and Policy Manager services (see Installing the Single Sign-On Service on page 117, Installing Reporting Service on page 129 and Installing Platform on page 135), install the Policy Manager service. The SSO service must be running when you run the Reporting Service installer.

To install the Policy Manager service:

1 Install the target J2EE application server. You can use the same J2EE server that hosts the SSO and/or the Policy Manager service.

2 Start the J2EE server hosting the SSO service. On JBoss, use the `SSO_HOME/bin/serverstart` script.

3 Execute the file `hp-soa-systinet-policymgr-2.52.jar`, located on the installation CD or in your distribution directory. In Windows, you can double-click on it in an exploration window. On all operating systems, it launches with the command **java -jar hp-soa-systinet-policymgr-2.52.jar**. Executing this file opens the installation wizard.

Note: On Windows networks, copy the installer jar to your local file system and run it from there instead of from the network.

To see the command-line options, run **java -jar hp-soa-systinet-policymgr-2.52.jar --help**. Use these options to set up and run a silent installation (see Using Silent Installation on page 149).

Important: If you are using SOA Systinet with a manually created database, you must run a silent installation.

4 The welcome screen opens with hardware and software requirements. Read this carefully before clicking **Next**, which opens the license page.

5 Read and accept the license. Click **Next** to proceed to the **Installation Folder** page.

6 Type or browse the path to your Policy Manager installation folder and click **Next**. The default path is `C:\Program Files\Hewlett-Packard\Systinet\PolicyMgr`. Each SOA Systinet component requires a separate installation subfolder. Click **Next** and the installer unpacks the distribution files to the chosen location.

7 Select the type of J2EE application server to which you are installing Policy Manager. Not all installation steps are performed for every application server. The subsequent installation path differs as shown in Table 13 on page 144.

Table 13. Policy Manager Installation Path by Application Server

Application Server	Description	Steps Performed in Default Installation
JBoss	By default, complete, automated deployment to the server is performed.	All (See Step 8.)
WebLogic, Oracle AS or WebSphere	A deployable EAR file is created. This must be deployed with application server tools.	Step 10, Step 11, Step 12, Step 13, Step 14, Step 20, Step 21, Step 22

8 Select either advanced or default deployment. Default deployment performs all steps that can be performed for the chosen application server. For advanced deployment, select a subset of steps to perform:

- Password Encryption, Step 9
- Database Setup, Step 10
- Configuration Table Setup, Step 11
- Endpoint Properties, Step 12
- SSO Identity Setup, Step 13
- Extensions Setup, Step 14
- Application Server Properties (JBoss only), Step 15

- Datasource Setup (JBoss only), Step 16
- SSL Setup (JBoss only), Step 17
- JMS Setup (JBoss only), Step 18
- Deployment (JBoss only), Step 19
- Platform Server Properties, Step 20
- Reporting Server Properties, Step 21

9 Select whether to encrypt passwords. Password encryption is recommended for production environments. If you set a passphrase for encrypting your passwords, include `-Dpassword.encryption.passphrase=`*passphrase* on the command line for non-interactive installation (Using Silent Installation on page 149) or the command-line tools that require authentication. To change the passphrase after installation, run the Setup tool in the **Advanced** scenario and select the **Password Encryption** option.

Caution: If you install more than one SOA Systinet component onto the same application server, so that they share the same db tablespace, you have the following passphrase limitations:

- You must use the same passphrase for all components sharing a tablespace.
- You cannot change the shared passphrase with the Setup tool. If you need to change a passphrase shared by multiple SOA Systinet components, contact HP professional services.

Do not use the administrator password as the passphrase.

If you later lose the passphrase, contact HP professional services for assistance.

10 Set up the database.

a Choose whether to create a new database, create a new schema in an existing database or configure an existing database and schema. If you are using the Setup tool, you can also drop a database or schema. Consult your database administrator for instructions. See Database Installation Types on page 27 for details.

b Select which type of database you are using, such as Oracle 10 or DB2.

c Type in database parameters. Please see Table 7 on page 29 for details. All SOA Systinet components must use the same database.

> **Note:** If you are using the Oracle database, you can enter the connection string instead of the server address, port and database name. You still must enter the user credentials.

d Type in or browse to the full paths of the JDBC driver .jar/.zip file to be installed to Policy Manager, separated by commas. For Oracle, this should be `ojdbc14.jar`. For DB2, these should be the two files `db2jcc.jar` and `db2jcc_license_cu.jar`. If you do not have these files, contact your database administrator.

e The installer now copies the JDBC drivers and verifies the connection to the database.

11 The installer now verifies that a configuration table is available.

12 Now specify the endpoint properties: Hostname, HTTP and HTTPS port numbers, whether or not to use HTTPS transport, and the web context. The default values are given for JBoss. If you are installing on another application server, type the port numbers of the managed server or cluster that will host Policy Manager.

If communication goes through a proxy server, use the proxy server's hostname and port numbers (see Proxy Setup on page 165). If SOA Systinet components should not use HTTPS to communicate between each other, unselect **Use HTTPS**.

> **On JBoss:** If you change the port numbers from their default values, you must also change the application server configuration to use these ports. See Configuring JBoss When SOA Systinet Uses Non-default Ports on page 56.

Only the specific hostname entered here during installation can be used with Policy Manager validation tools, either in the UI or on the command line. For example, if you enter the absolute address of your local machine as the hostname, you must use that absolute address and not "localhost" with validation tools.

13 Type in the following information:

SSO Service URL	The URL of the SSO Service. If SOA Systinet components communicate through a proxy server, use the proxy server's URL. (If a proxy server is used for external communication but SOA Systinet components communicate with each other directly, use the SSO service host's URL.)
Configuration Service Admin Name	Name and password of the SSO Service administrator. The default password is `changeit`.
Configuration Service Admin Password	
Identity Name	Arbitrary name of the identity for the Policy Manager service you are creating on the SSO Server. Default is `policymgr`.
Identity Password	The password of the SSO identity you are creating. The default password is `changeit`.

14 The installer now validates that the extensions can be applied.

If you are installing on a WebLogic, Oracle or WebSphere application server, go to Step 20.

15 (JBoss only) Indicate path and configuration type of JBoss application server. If installing Policy Manager standalone, the configuration should be `default` or point to a copy you made of `JBOSS_HOME/server/default`. If installing Policy Manager to a cluster, the configuration should be `all` or point to a copy you made of `JBOSS_HOME/server/all`.

16 (JBoss only) The installer now verifies the datasource.

17 (JBoss only) The installer now verifies the existence of the necessary application server configuration files for SSL communication.

18 (JBoss only) The installer now verifies JMS setup.

19 (JBoss only) The installer now verifies application server properties.

20 In the **Platform SSO Partner** panel, enter the SSO partner identity name of the platform component that you are linking to this Policy Manager component. First select whether to choose from a list of known SSO identities or type in a partner name manually. Then, if you are selecting from the list, highlight the selected identity name. The default partner identity name you want is **platform**.

21 In the **Reporting Server Properties** panel, select the SSO partner identity name of the reporting service component that you are linking to this Policy Manager component. The default partner identity name you want is **reporting**. You can also choose to use the secure SSO https URL .

22 Depending on the type of installation, the installer either creates a deployable EAR file in `POLICYMGR_HOME/deploy` or deploys Policy Manager to the JBoss application server.

15 Using Silent Installation

You can use SOA Systinet installer command-line options to install SOA Systinet components non-interactively or install them using a reuseable installation configuration. The installation wizard generates an XML property file with the installer configuration. This file can be edited and used in a non-interactive installation.

To see a list of all installation options and a brief description of each, run **java -jar *`installer_jar_file`* --help**. These options are the same as those for the Setup tool, described in the Administration Utilities part of the Administrator Guide.

To install SOA Systinet non-interactively:

1. For each component of SOA Systinet, create a configuration file with the SOA Systinet component's installation GUI without installing the component. To run the installer without installing, use the `--save-config | -s` option. For example, to create a configuration file for installing SSO service, named `sso-config.xml`, run **java -jar hp-soa-systinet-sso-2.52.jar -s sso-config.xml**.

2. While creating the configuration file, you unpacked the installation JAR to an installation folder you specified in the GUI. If you want the installer to unpack the installation JAR to that same location when you run silent installation (see Step 3), delete the unpacked installation folder. The installer cannot overwrite the unpacked installation folder.

3. Run silent installation using the configuration file you created in Step 1. Specify the directory to which the installer will unpack the installation JAR file. For example, to install SSO service silently using the configuration file `sso-config.xml` and with SSO_HOME at `/opt/hp/systinet/sso`, run **java -jar hp-soa-systinet-sso-2.52.jar -u sso-config.xml -i /opt/hp/systinet/sso -q**.

   If you encrypt passwords for SOA Systinet, include the `-Dpassword.encryption.passphrase=XXX` switch when you run silent installation, for example **java -Dpassword.encryption.passphrase=myPassPhrase -jar hp-soa-systinet-sso-2.52.jar -u sso-config.xml -i /opt/hp/systinet/sso -q**.

Part V. After Installation

This part describe procedures you run after installing SOA Systinet. It contains the following chapters:

- Self-Testers on page 153. A Platform tool that automatically checks the setting of critical J2EE resources and parameters supplied by an application server and displays the results in a server console and a browser window. It can be deployed independently of SOA Systinet Platform.
- Enabling Full Text Search on page 157. How to set up the SOA Systinet Full Text Search feature.
- Importing SOA Systinet Registry Certificate on page 163. How to integrate with a legacy installation of HP SOA Systinet Registry.
- Proxy Setup on page 165. How to set up both proxy servers and SOA Systinet so that SOA Systinet communicates through a proxy.

16 Self-Testers

SOA Systinet Platform, Policy Manager and Reporting each include a self tester that checks the setting of J2EE resources and parameters supplied by an application server. Each self tester runs automatically every time its associated component is started. When the bundled Platform self tester does not function, you can deploy the Platform self tester as a standalone application.

The self testers check the following:

Table 14. Self-Tests By Component

Self-Test	Description		SOA Systinet Component
JNDI	Checks that correct resources are configured and their JNDI names are properly spelled.		All
Datasource	• Checks that datasource can be connected and data retrieved. • Checks that the component's endpoint URLs are defined and prints the URLs to the output.		All
JMS	Sends testing message to all queues and topic. The user can see in the server console whether the test messages have been received by message listeners. (The console displays listener log info about the test message.)		All
SSL	Platform can turn off certificate verification. The self tester checks whether certificate verification is enabled or disabled.		Platform
SSO checkers	SSOChecker	Checks that a particular SOA Systinet component is configured with SSO.	All
	SSOAvailabilityChecker	Checks that the SSO that is configured for the product is running.	
	SSOPartnerChecker	Checks that the product is registered in SSO.	
	SSODescriptorContentsChecker	Checks the results of registration of the component to SSO. The base URLs defined for the component (locally) must match the URLs that are registered for that product in SSO.	

Self-Test	Description	SOA Systinet Component
Other checks	• Checks for presence of log4j, prints the log4j configuration file location to the console. • Prints out system properties.	All
Application server-specific	• WebSphere — Tests work manager existence. • Oracle AS — Tests oc4j.userThreads system property setting.	Platform
ConfigurationChecker	Checks for existence of mandatory configuration properties and checks the validity of URLs.	Policy Manager
PartnerChecker	Tries to connect to Reporting and SSO. Checks whether the descriptors of SSO, Reporting and Platform exist and compares the registered URLs with the URLs defined in the configuration.	Policy Manager
ExtensionChecker	Checks whether Policy Manager's extensions are installed into Platform. Tries to get collections of assertions, business policies and tools.	Policy Manager
Missing JARs	Checks for missing JAR files (xalan, xerces, common-logging, dom4j, hibernate, etc.) necessary to Policy Manager.	Policy Manager
ResourceFeedChecker	Validates the reporting/reports and reporting/libraries resource feeds, in two steps: 1 Parses feed XML and checks if root element is <feed>. Shows error message if there is a problem. 2 Checks that at least one entry is present. If not, it shows a warning.	

Output of the self tester can be read either in the component's server console or as web output. In the default configuration, the server console output includes only information about groups of checks that are run and any errors that occur. The web output is more informative, showing all checks that are performed. It is also

designed for better readability. The web output is accessible at the following URLs (one for each component's self tester):

- `http://hostname:port/soa/self-test`
- `http://hostname:port/policymgr/self-test`
- `http://hostname:port/reporting/self-test`

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

Independent deployment (Platform-only) . In cases where the self tester bundled with Platform cannot be accessed, such as when the server is badly configured and Platform cannot be deployed, the user can deploy the Platform self tester as a standalone application. The package is prepared for deployment in `PLATFORM_HOME/deploy/self-test.war`. The web output from the standalone self tester is accessible at `http://hostname:port/self-test/self-test` (Note that the self-test context can be changed in most application servers.)

You can test connections in the self-test web output UI. Click the **Test** link next to the URL link that you want to test (for example, a link to the SSO descriptor). The test simulates the creation of a connection from within SOA Systinet. This simulation is not otherwise possible. It also helps with the setup of SSL by printing out the certificates that are required to be trusted when the connection fails because of SSL.

Important: The self tester's web output is accessible to everyone. For security reasons, switch off the self tester web output after a successfully completed installation passes the self-test.

To switch off web output for the bundled self tester:

1. Log on to SOA Systinet Platform as administrator.
2. Open the **Tools** tab and go to **Administration->Configuration**.
3. Unselect **Self Test Access**.

To disable the Platform standalone self tester, undeploy the `self-test.war` package from your server. Go to the self-test web url and verify that the self-test has been disabled.

17 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. Optionally disable the appendage of **%** characters to search terms (useful if using non-Latin character sets).
3. Activate FTS in the SOA Systinet UI as described in the SOA Systinet Configuration Options section in the HP SOA Systinet Administrator Guide.

The following sections provide details on enabling Full Text Search:

- Creating Full Text Search Indexes on DB2 on page 158
- Creating Full Text Search Indexes on Oracle on page 159
- Disabling the Addition of % to Search Terms on page 160

Ensure that your database server meets the system requirements described in the Database Setup Overview on page 25.

Creating Full Text Search Indexes on DB2

To 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 1 on page 158.

Example 1: 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>
```

Example 2 on page 158 are commands you use to update the index manually.

Example 2: 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 also DB2 Net Search Extender documentation.

Creating Full Text Search Indexes on Oracle

In order to create indexes and schedule their update, use the Oracle **sqlplus** console. Connect to the database using the same credentials used during installation.

Example 3 on page 159 shows the procedure in commands. 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 3: 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")');

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")');
```

When you create the index, remove words that could frequently appear in full-text searches from the Oracle stoplist. By default, the Oracle index stoplist includes words such as "to." Full-text searches including these words return a false empty list. Alternatively, the database administrator should provide SOA Systinet users with the stoplist and a warning not to use these terms in full-text searches. Example 4 on page 160 is a set of commands for replacing the Oracle stoplist.

Example 4 on page 160 shows the commands to set up the indexing stoplist on Oracle:

Example 4: Creating Oracle Indexing Stoplist

```
call CTX_DDL.CREATE_STOPLIST('MyStoplist');
call CTX_DDL.ADD_STOPWORD('MyStoplist', 'a');

... Add words that should not be indexed. Do not include "to" or other words that would frequently
        occur in full-text searches.

-- 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")') ;
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")');
```

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

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). You can use any other synchronization technique.

Example 5 on page 160 shows how to execute index synchronization manually.

Example 5: 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');
```

Disabling the Addition of % to Search Terms

By default, full-text search in SOA Systinet adds a % to the end of input search terms. This functionality might cause problems when using non-Latin character sets.

To disable the addition of %:

1. Stop SOA Systinet. If using JBoss, stop SOA Systinet with the `PLATFORM_HOME/bin/serverstop` command.
2. Export the configuration with the command:

   PLATFORM_HOME/bin/configurationexport config.xml
3. Open `config.xml` for editing and search for this element:

```
<fulltextsearch>
    <appendpercentage>true</appendpercentage>
</fulltextsearch>
```

4. Change the value of `appendpercentage` to `false`, and save the file.
5. Import the amended configuration with the command:

   PLATFORM_HOME/bin/configurationimport config.xml
6. Start SOA Systinet. If using JBoss, start SOA Systinet with the `PLATFORM_HOME/bin/serverstart` command.

18 Importing SOA Systinet Registry Certificate

If you are using SOA Systinet in conjuction with HP SOA Systinet Registry and plan to import artifacts and taxonomies, you need to import the HP SOA Systinet Registry certificate. The way you do this varies according to the application server on which you have deployed SOA Systinet as described in the following sections:

- Importing Systinet Registry Certificate to JBoss on page 163
- Importing Systinet Registry Certificate to WebLogic on page 164
- Importing Systinet Registry Certificate to WebSphere on page 164
- Importing Systinet Registry Certificate to Oracle Application Server on page 164

Importing Systinet Registry Certificate to JBoss

If HP SOA Systinet Registry is installed standalone, it has a self-signed certificate. To import the HP SOA Systinet Registry certificate to SOA Systinet deployed on JBoss, run the following command (You need the SOA Systinet Platform truststore password, which is `changeit` by default):

keytool -import -alias registry -file "C:\Program Files\Hewlett-Packard\Systinet\registry\doc\registry.crt" -keystore "C:\Program Files\Hewlett-Packard\Systinet\platform\conf\client.truststore"

In the syntax of that command, SOA Systinet and HP SOA Systinet Registry are installed to the default locations on Windows machines. HP SOA Systinet Registry is installed standalone and uses its default self-signed certificate. If SOA Systinet or HP SOA Systinet Registry are installed to different locations, modify the command accordingly. If HP SOA Systinet Registry is installed to an application server, it uses a certificate in that application server's truststore. Export the certificate to a CRT file before importing it with the **keytool -import** command above.

Importing Systinet Registry Certificate to WebLogic

Use the Sun Java `keytool` or the WebLogic console to create a truststore for the server hosting SOA Systinet Platform or its proxy. Then import the HP SOA Systinet Registry self-signed certificate or host server's certificate to the trust store you created.

Importing Systinet Registry Certificate to WebSphere

Use IKEYMAN or the WebSphere console to create a truststore for the server hosting SOA Systinet Platform or its proxy. Then import the HP SOA Systinet Registry self-signed certificate or host server's certificate to the trust store you created.

Importing Systinet Registry Certificate to Oracle Application Server

If HP SOA Systinet Registry is installed standalone, it has a self-signed certificate at `REGISTRY_HOME\doc\registry.crt`. If it is installed on an application server, its certificate location depends on how that application server handles SSL. In either case, import the HP SOA Systinet Registry certificate to the truststore of the OAS hosting SOA Systinet Platform. Follow the instructions for trusting a CA certificate given in Setting Up SSL on OAS on page 102.

19 Proxy Setup

SOA Systinet can communicate via a proxy server. This enhances and simplifies security, allows all external communication through a single URL and relieves the need for SSL communication between SOA Systinet components. Communication via a proxy server can be configured in one of these ways:

- The components of SOA Systinet communicate with each other via the proxy server.
- The components of SOA Systinet communicate with each other directly. Only external communication goes through the proxy.

Setup is identical for both cases This applies whether or not you use HTTPS for communication between SOA Systinet components and the proxy. HTTPS vs HTTP is chosen when you specify the component endpoint in each SOA Systinet installer (see for example Step 12 in Installing the Single Sign-On Service on page 117).

To set up SOA Systinet when using a proxy server:

1. Prepare and configure the proxy server. Provide the following URL mappings at the proxy (all URLs end with the service context, such as /sso, /platform etc.):
   - [PROXY_SSO_URL] to [SSO_HOST_URL]
   - [PROXY_PLATFORM_URL] to [PLATFORM_HOST_URL]
   - [PROXY_REPORTING_URL] to [REPORTING_HOST_URL]
   - [PROXY_POLICYMGR_URL] to [POLICYMGR_HOST_URL]

     Provide reversed mapping that rewrites 'Location' and 'Content Location' from [POLICYMGR_HOST_URL] to [PROXY_POLICYMGR_URL]/policymgr.

   If you intend to use secure communication protocol for Web UI between public clients and the proxy — enabled by 'Use HTTPS' option in the installation wizzards — then also configure SSL on proxy.

2 Follow the standard setup and installation procedure for your application server described in Part III, "Deploying SOA Systinet To Application Servers" until you have completed deploying SSO service. Be certain to use the proxy's hostname and port number when you specify the endpoint properties (see Step 12 in Installing the Single Sign-On Service on page 117).

3 Add the URL of the server or cluster hosting SSO service to the SSO partner.

a On the command line, type `SSO_HOME/bin/ssoconfig.bat|.sh get -e urn:systinet2:sso:1.2:[PROXY_HOSTNAME]:[PROXY_SSL_PORT_NUMBER] -d . -s [SSO DMZ URL]`. In this command, [SSO DMZ URL] is the URL of the server or cluster hosting SSO service and the string beginning `urn:systinet2...` is the SSO partner ID. The command creates a file named `[SSO_PARTNER_ID].xml` in your current location.

b Open the file `[SSO_PARTNER_ID].xml` and find the following element:

```
<urn:EntityDescriptor...>
...
   <urn:IDPSSODescriptor...>
    ...
      <urn:Extensions...>
```

c Add the following subelement to the `urn:Extensions` element: `<ext:BaseUrl ext:pathPrefix="">[SSO DMZ URL]</ext:BaseUrl>`

d In the command line, run this command: `SSO_HOME/bin/ssoconfig.bat update -f [SSO PARTNER ID].xml -s [SSO DMZ URL] -u username -p password` . The default username is `admin` and the default password is `changeit`. This command tells SSO service to use the configuration you modified.

4 Complete the standard setup and deployment procedure for your application server.

5 *If SOA Systinet components communicate with each other directly*, set up redirection:

a Copy the following configuration file to your file system with the name `config.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>

<properties>
    <shared>
        <url>
            <redirection>
                <from-0>[PROXY_URL]/soa</from-0>
```

```
                <to-0>[PLATFORM_HOST_URL]/soa</to-0>
                <from-1>[PROXY_URL]/reporting</from-1>
                <to-1>[REPORTING_HOST_URL/reporting</to-1>
                <from-2>[PROXY_URL]/sso</from-2>
                <to-2>[SSO_HOST_URL]/sso</to-2>
  <!-- #3 applies to standard editions only -->
                <from-3>[PROXY_URL]/policymgr</from-3>
                <to-3>[POLICYMGR_HOST_URL/policymgr</to-3>
            </redirection>
        </url>
    </shared>
</properties>
```

b Replace the variables in the file with real URLs. Save when done.

c In the command line, run this command: `PLATFORM_HOME/bin/configurationimport.bat|.sh config.xml`

d Restart all host servers.

Index

A

Active directory
  installation, 121
AIX
  supported platforms, 17
application server
  WebLogic, 59
  WebSphere, 81
application servers
  recommended OS, 17
authentication
  LDAP, 126

C

certificate
  LDAP, 127
certificates
  JBoss, 54
  WebLogic, 65
  WebSphere, 82
connection factory
  WebLogic, 71
connection pool
  JBoss, 55
  WebLogic, 67
  WebSphere, 83

D

database
  creation, 27
  operations, 27
  schema, 27
  tablespace, 27
datasource
  WebLogic, 67
  WebSphere, 83
DB2
  setup, 25
  supported platforms, 17
domain
  WebLogic, 59

E

external accounts, 121
  LDAP, 121

F

full text search
  setup, 157

H

hardware
  requirements, 15
hostname verification error
  LDAP, 126
HP-UX
  supported platforms, 17

I

installation
  Active directory, 121
  external accounts, 121
  LDAP, 121
  Policy Manager, 143
  Reporting Service, 129
  SOA Systinet Platform, 135

SSO, 117

J

J2EE
WebLogic, 59
WebSphere, 81
J2EE platforms
with recommended OS, 17
Java
requirements, 15
JBoss
certificates, 54
connection pool, 55
deploying SOA Systinet, 41
JMS, 49
memory allocation, 55
Policy Manager installation, 143
ports,
Reporting Service installation, 129
run script,
setup, 41
Single Sign-on, 117
SOA Systinet Platform installation, 135
SSL, 54
supported platforms, 17
JDBC
WebLogic, 67
WebSphere, 83
JMS
JBoss, 49
WebLogic, 70
WebSphere, 81
JMS Modules
WebLogic, 71
JMS queue
WebLogic, 71
JMS Servers
WebLogic, 71
JMS Topic
WebLogic, 71
JSSE
LDAP, 127

K

keystore
WebLogic, 65
WebSphere, 82
keytool
LDAP server trust, 127

L

LDAP
installation, 121
SSL, 126
TLS, 126
ldaps, 126
linux
supported platforms, 17
log4j
WebLogic, 59
WebSphere, 89

M

mail session
WebLogic, 77
WebSphere, 83
memory allocation
JBoss, 55
WebLogic, 59
WebSphere, 89
message store
WebLogic, 67
WebSphere, 83

O

operating system
  recomended application servers, 17
Oracle
  setup, 25
  supported platforms, 17
Oracle Application Server (OAS)
  Policy Manager installation, 143
  Reporting Service installation, 129
  Single Sign-On, 117
  SOA Systinet Platform installation, 135

P

PLATFORM_HOME, 135
Policy Manager
  installation, 143
Policy Manager installation
  JBoss, 143
  Oracle Application Server (OAS), 143
  WebLogic, 143
  WebSphere, 143
POLICYMGR_HOME, 143
profile
  WebSphere, 82

R

REPORTING_HOME, 129
Reporting Service
  installation, 129
Reporting Service installation
  JBoss, 129
  Oracle Application Server (OAS), 129
  WebLogic, 129
  WebSphere, 129
requirements, 15

S

Single Sign-On (SSO)
  JBoss, 117
  Oracle Application Server (OAS), 117
  WebLogic, 117
  WebSphere, 117
SOA Systinet Platform
  installation, 135
SOA Systinet Platform installation
  JBoss, 135
  Oracle Application Server (OAS), 135
  WebLogic, 135
  WebSphere, 135
software
  requirements, 15
SSL
  JBoss, 54
  LDAP, 126
  WebLogic, 65
  WebSphere, 82
SSO_HOME, 117
supported platforms
  recomended application servers and OS, 17
system property
  LDAP, 127

T

trust
  LDAP, 127
  WebLogic, 65
  WebSphere, 82

U

UNIX
  supported platforms, 17

W

WebLogic, 59
- connection factory, 71
- connection pool, 67
- datasource, 67
- domain, 59
- JDBC, 67
- JMS, 70
- JMS Modules, 71
- JMS Queue, 71
- JMS Servers, 71
- JMS Topic, 71
- log4j, 59
- logging, 59
- mail session, 77
- memory allocation, 59
- message store, 67
- Policy Manager installation, 143
- Reporting Service installation, 129
- Single Sign-on, 117
- SOA Systinet Platform installation, 135
- SSL, 65
- supported platforms, 17
- trust, 65

WebSphere, 81
- connection pool, 83
- datasource, 83
- JDBC, 83
- JMS, 81
- log4j, 89
- logging, 89
- mail session, 83
- memory allocation, 89
- message store, 83
- Oracle setup, 25
- Policy Manager installation, 143
- profile, 82
- Reporting Service installation, 129
- Single Sign-on, 117
- SOA Systinet Platform installation, 135
- SSL, 82
- supported platforms, 17
- trust, 82

Windows
- supported platforms, 17