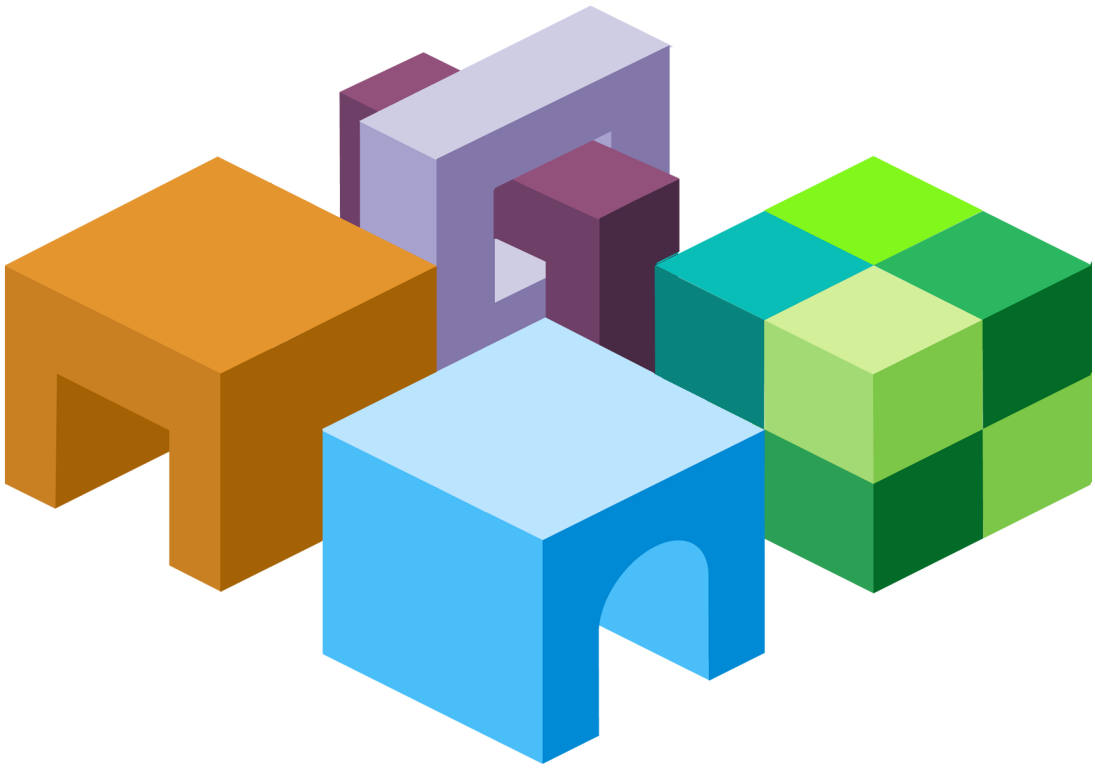


# HYPERION® SHARED SERVICES

*RELEASE 9.3.1.0.03*

## INSTALLATION GUIDE FOR UNIX

ORACLE | Hyperion

P/N: DH98593100

Shared Services Installation Guide for UNIX, 9.3.1.0.03



Authors: Michelle Cohen, Shannon Murray

# Contents

# 1 Overview

**In This Chapter**



## Shared Services Introduction

**Note:** This book is specific to the Oracle's Hyperion® Shared Services 9.3.1.0.03 Service Fix and replaces the original 9.3.1.0 installation guide. For information about the issues addressed in the Service Fix, see the Shared Services 9.3.1.0.03 Readme.

Shared Services functionality is programmed into products, such as Oracle's Hyperion® Planning – System 9, Oracle's Hyperion® Financial Management – System 9, Oracle's Hyperion® Business Modeling, and Oracle's Hyperion® Performance Scorecard – System 9. Shared Services integrates the products to provide these functionalities:

- User provisioning
- External authentication definition
- Task flow management

The *Hyperion Security Administration Guide* describes user-provisioning functionality and external authentication definition. All other Shared Services functionality is described in the administrator's and user's guides for the products that implement Shared Services. Products that implement Shared Services functionality require access to a Shared Services server running Shared Services client and server software, and to a database dedicated to Shared Services.

## Shared Services Components

- Shared Services Server
- Shared Services Documentation
- Hyperion Security Platform
- Oracle's Hyperion® Shared Services User Management Console

**Note:** Hyperion no longer ships or requires Oracle's Hyperion® License Server™ (or standalone license files) for use with Hyperion products. To ensure compliance with your license agreement, Hyperion recommends that you implement an auditing process. In addition, during product configuration with Oracle's Hyperion® Configuration Utility™, you activate only the features you purchased. For more information, see "Hyperion License Compliance" in *Hyperion Installation Start Here*.

## Shared Services Server

The Shared Services server components:

- Databases (relational and OpenLDAP)
- Web application server
- User Management Console

You only need to install the Shared Services server to one computer and the server does not need to reside on the same computer as products registering with Shared Services. Descriptions of the Shared Services server components follow.

### Databases

Shared Services stores its data in two databases:

- An OpenLDAP database

  The OpenLDAP database stores native Shared Services users.

- A relational database

  The relational database stores the event and administrator-related data. For a list of supported relational databases, see the *Hyperion Installation Start Here*.

For all supported database software, the installation installs the required JDBC drivers.

Following Shared Services installation, you must complete the configuration process as specified in Chapter 3, "Configuring Shared Services."

### Application Servers

Shared Services requires an application server and you must install Shared Services to the same computer running the application server. For a list of supported application servers, see the *Hyperion Installation Start Here*.

**Note:** Hyperion provides Apache Tomcat on the installation media for convenience if you want to use it for your deployment. Hyperion does not own or maintain the Apache Tomcat application server and is not responsible for problems you may encounter with its functionality. Hyperion, however, fully supports the use of Apache Tomcat in its products. In deployments where customers require high availability or failover, Hyperion recommends you deploy a commercially supported application server where these capabilities are supported.

Following Shared Services installation, you can either deploy the application server automatically using the Hyperion Configuration Utility (see Chapter 3, "Configuring Shared Services") or you can manually configure the application server for use with Shared Services. For manual configuration instructions, see one of the following manual deployment appendices:

- Appendix A, "Manual Deployment to WebLogic Application Server"
- Appendix B, "Manual Deployment to WebSphere Application Server"
- Appendix C, "Manual Deployment to Oracle Application Server"

**Note:** You cannot automatically configure Oracle Application Server for use with Shared Services.

### User Management Console

User Management Console is the centralized user interface for these tasks:

- Managing projects and applications within projects
- Provisioning users and groups for applications
- Managing the Shared Services native directory

The console software is installed with Shared Services server. See the *Hyperion Security Administration Guide*.

## Shared Services Documentation

For a list of Hyperion documents related to Shared Services, see the Hyperion Shared Services Information Map. The Information Map is available from the User Management Console Help menu for all operating systems.

All Shared Services documentation is accessible from the following locations:

- Oracle's E-Delivery Web site (http://edelivery.oracle.com/)
- The product DVD
- The Information Map
- Online help, available from within the User Management Console

  After you log on, you can access online help by clicking Help or selecting the Help menu.

## Hyperion Security Platform

The Hyperion security platform is a framework enabling Hyperion applications to use external authentication and single sign-on. External authentication means the user login information needed by Hyperion applications is stored outside the applications in a central authentication directory. Single sign-on is the ability of a user to access multiple Hyperion products after logging on only once. For detailed information about Hyperion security platform, see the *Hyperion Security Administration Guide*.

## Shared Services User Management

Shared Services user management enables centralized management of Hyperion product users. The user management system enables administrators to group applications into projects, and then provision users and groups with roles and access rights for applications.

You can provision users and groups whose accounts are maintained in corporate authentication directories and in the native directory. Each Hyperion product provides product-specific roles, and Shared Services provides system-wide roles.

For detailed information about Shared Services user management, see the *Hyperion Security Administration Guide*.

# Planning the Shared Services Installation

## Preparing to Install Shared Services

Before installing Shared Services, note the following database requirements:

- If you are using an IBM DB2 database, increase the `applheapsz DB2` configuration parameter to `4000`. Use this command:

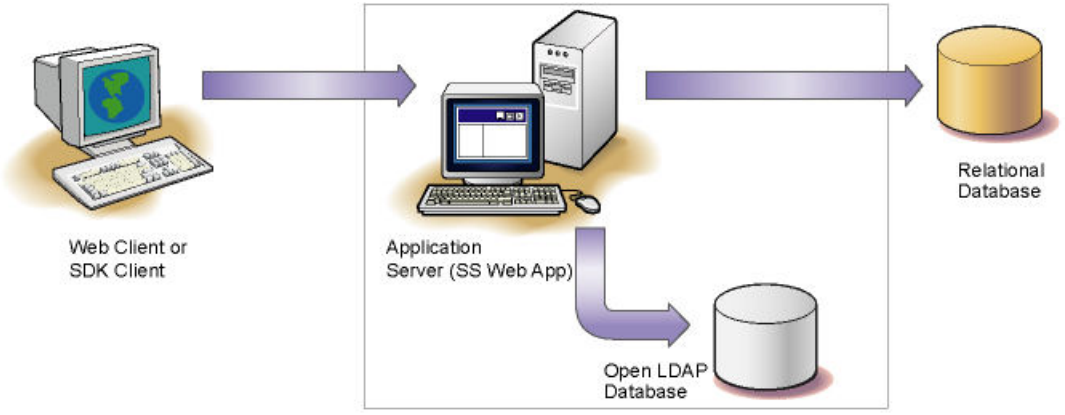
  ```
  update db cfg for HUB_DB_NAME using applheapsz 4000
  ```

  This parameter becomes effective when the IBM DB2 instance is restarted.
- Shared Services requires the UTF-8 character set for all databases. For information about installing the database to support UTF-8, see the database documentation.
- Do not run an OpenLDAP database on a Network File System (NFS) mounted protocol.

## Shared Services Deployment Scenarios

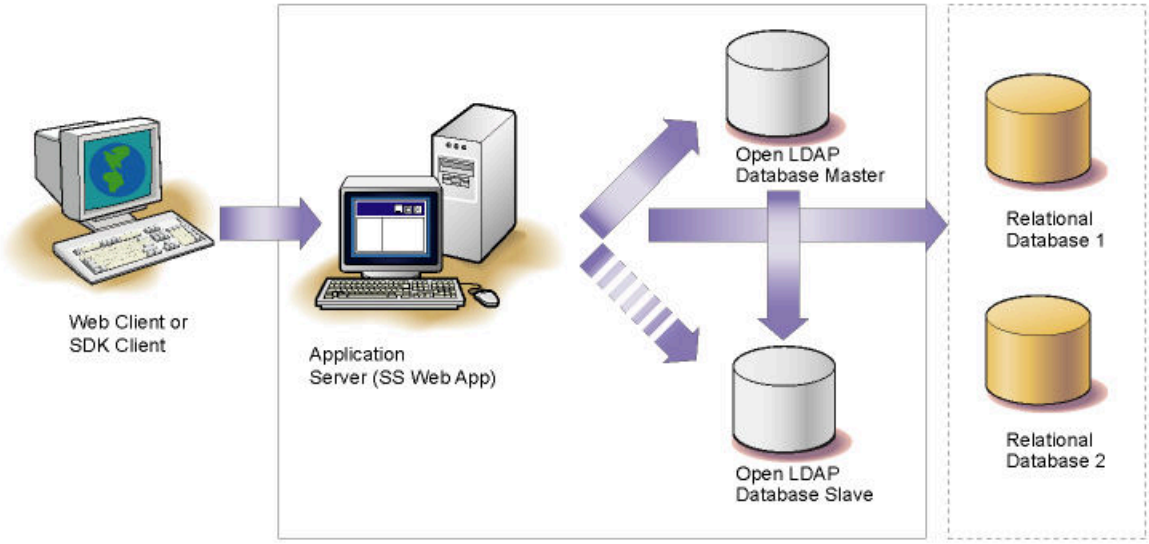
### Standard Configuration

This scenario depicts the default configuration of Shared Services: one instance of Shared Services Web application, one instance of the OpenLDAP database, and one instance of a relational database. This scenario does not support failover or load balancing.




## Configuration with Replicated Databases

This scenario deploys Shared Services on only one application server but provides failover support for the database.




This scenario works for deployments that exhibit these characteristics:

- Authentication to Hyperion products is a critical issue.
- Authentication needs failover support.
- Features such as user provisioning and taskflow management do not need failover support.

Relational database replication and failover is vendor-specific. Shared Services provides the ability to replicate OpenLDAP and configure it to support failover. See Appendix D, "Shared Services Backup and Recovery."

**Note:** There is no load balancing support for OpenLDAP.

# 2 Installing Shared Services

**In This Chapter**



## What Happens During Installation

During installation, the Shared Services installer program performs the following operations:

- Creates directories and subdirectories under the location specified during installation

  In documentation, the directory to which Shared Services is installed is called *<HSS_HOME>*. For a list of the Shared Services directories created during installation, see "Files Installed in the HSS_HOME Directory" on page 14.

- Installs Hyperion common components to *<HYPERION_HOME>*/common

  For information about *<HYPERION_HOME>* and a list of directories created, see "About Hyperion Home" on page 14.

  After *<HYPERION_HOME>* is defined, you can run a migration utility to change its location. Hyperion Home Migration Utility is provided with the Shared Services installation.

- A typical Shared Services installation installs these components:
  - OpenLDAP database (stores the security-services-related data)
  - Shared Services server
  - Shared Services documentation
- A custom Shared Services installation installs only the components that you select.

  **Note:** You must install and configure a supported database. For a list of supported relational databases, see the *Hyperion Installation Start Here*.

- Installs documentation files to *<HYPERION_HOME>*/common/docs/SharedServices on the Shared Services computer

  See "Shared Services Documentation" on page 9.

- Installs an uninstaller in <*HSS_HOME*>/uninstall

  See Chapter 4, "Uninstalling Shared Services."

## Files Installed in the HSS_HOME Directory

Shared Services files are installed in folders within the <*HSS_HOME*> directory, by default, /home/*username*/Hyperion/SharedServices/<*releaseNumber*>

**Table 1 Shared Services Folders Created for Installations**

| Folder | Contents |
|---|---|
| AppServer | InstallableApps—war and ear files and scripts and other files needed to deploy an application; as part of preparing a war or ear file for deployment, the war file MAY be unpacked here and some files modified, then the war or ear file repacked, ready for deployment<br>InstalledApps (for WebLogic 8.1.*x* only)—folder created by Hyperion Configuration Utility during an automatic deployment of an application; the war file contained within is sourced from InstallableApps and deployed for the user |
| client | Java APIs for applications to use Shared Services functionality |
| openLDAP | OpenLDAP database files |
| server | Shared Services executable files, default relational database files, Java class files, and server locale files |
| uninstall | Files for uninstalling Shared Services |

## Installation of JDK on AIX

For AIX, the Shared Services installer does not perform a full installation of the JDK. Instead, the Java files required for you to install JDK are placed in the <*HYPERION_HOME*>/common/JDK/IBM/1.4.2 directory:

- fixes.html
- java14.license
- java14.sdk
- sdkguide.aix32.htm

See sdkguide.aix32.htm for instructions on installing the JDK.

# About Hyperion Home

When multiple Hyperion products are installed on one computer, common internal and third-party components are installed to a central location, called *Hyperion Home*. The Hyperion Home value is stored in .hyperion.<hostname> in the home directory.

**Note:** To ensure that all installers have the permissions required to modify the *HYPERION_HOME* location, Hyperion recommends that all Hyperion applications be installed under one HYPERION user account.

## Hyperion Home Location

The default location for Hyperion Home is `$HOME/Hyperion`. When you install, the installer searches for the `HYPERION_HOME` environment variable on the computer to which you are installing.

If the Hyperion Home location was previously defined for another Hyperion product, the installation uses the previously defined location. The location cannot be changed through the installer.

If the current installation is the first Hyperion installation on the computer, you can specify the location during installation.

**Note:** If the HYPERION_HOME directory is mounted on an NFS so that one HYPERION_HOME location is visible across multiple computers, Shared Services can be installed to only one computer. If you try to install Shared Services to an additional computer, the previous installation is detected.

## Files Installed in the HYPERION_HOME Directory

Various files are installed in the `HYPERION_HOME/common` directory by a default installation of Shared Services. Some common components, and thus some files and folders, are optional and may not be installed.

**Table 2** Common-Component Folders Created in the Common Directory

| Folder | Contents |
|---|---|
| `appServers` | Application server files |
| `CLS` | License services APIs |
| `config` | Hyperion Configuration Utility files |
| `CSS` | Files to supportHyperion external authentication |
| `Docs` | Product documentation files |
| `EssbaseJavaAPI` | Java driver used when embedding Essbase in other applications |
| `EssbaseRTC` | Essbase runtime client used when embedding Essbase in other applications |
| `httpServers` | Apache web server files for batteries included installation |
| `HyperionLookAndFeel` | Installer user interface files |
| `JakartaCommons` | Common development library files |

| Folder | Contents |
|---|---|
| JavaMail | Files to support sending e-mail via Java |
| JCE | JCE files for encryption, key generation and agreement, and MAC |
| JDBC | JDBC files |
| JRE | Java Runtime Environment files |
| lib | common internal library files |
| loggers | Files for external authentication logging |
| ODBC | ODBC drivers |
| Opatch | Oracle patching tool files; for future use |
| PERL | Scripting language files |
| SAP | SAP files |
| SharedServices | Supporting files for Shared Services |
| utilities | Utilities to change the location of Hyperion Home, and export, import, or validate provisioning data, or to support the promotion of artifacts across product environments and operating systems (for Oracle's Hyperion® Reporting and Analysis – System 9) |
| validation | Not used in this release |
| velocity | Not used in this release |
| XML | Common XML components |

## Changing the Hyperion Home Location

After Hyperion Home is defined through the product installation, you can run a migration utility to change the Hyperion Home location.

The migration utility updates the `.hyperion.<HOSTNAME>` file, which resides in the directory that contains the environment variable. Login initialization files, such as `.profile` and `.login` are not updated.

Hyperion Home Migration Utility is provided with the Shared Services installation.

➤ To change the Hyperion Home location:

1 **Launch the migration utility:**

- Choose a method:
  - In XWindows, change to `<HYPERION_HOME>/common/utilities/HyperionHomeTool/9.3.1/bin`. Then type `migrationtool.sh`.
  - In a UNIX console, change to `<HYPERION_HOME>/common/utilities/HyperionHomeTool/9.3.1/bin`. Then type `migrationtool.sh -console`.

2 **Step through the screens, and when prompted, enter the Hyperion Home location or click Browse to navigate to the preferred location.**

Do not choose a `HYPERION_HOME` location that contains a space character. For example, `$HOME/Program Files` is not acceptable.

# Launching Installers

➤ To launch the Shared Services installer:

1 **Download the installer from Oracle's E-Delivery Web site.**

2 **Extract the files from the ZIP file.**

After extraction, the installation directory should include the following files:

- `setup.bin`
- `setup.jar`

3 **Run the install setup program,** `setup.bin`.

**Note:** If you cannot execute `setup.bin`, use `chmod` to add execute privilege.

# Installing Shared Services

## Running the Installation Wizard

➤ To run the installation wizard to install Shared Services:

1 **Verify that all preparation tasks are complete and system requirements are met.**

See *Hyperion Installation Start Here*.

2 **Download and launch the Shared Services installer program.**

See "Launching Installers" on page 17.

3 **Follow the installation prompts, remembering this information:**

- If the installation wizard detects a previous installation of Shared Services, you must install the Shared Services files to the previous-installation directory.
- If no previous installation of Shared Services is detected, you can specify the directory in which to install the Shared Services files.
- Directory names are restricted in several ways:
  - You can enter only English alphanumeric characters and these special characters:

    dash ( - ), underscore ( _ ), backslash ( \ ), forward slash ( / ), dot (.)

    - You cannot install Shared Services files on a WebLogic or WebSphere application server to a directory whose name contains a space. For example, `$HOME/Program Files` is not acceptable.
- By default, the wizard installs Shared Services in `/home/username/Hyperion/SharedServices/<releaseNumber>`.
- Hyperion common components and Shared Services software must be installed to different directories. Common components are installed to a location called "Hyperion Home" (`<HYPERION_HOME>/common`). If the installation wizard detects an existing Hyperion Home setting, common components will be installed to that location. See "About Hyperion Home" on page 14.

  **Note:** If the `<HYPERION_HOME>` directory is mounted on a network file system (NFS) so that one `<HYPERION_HOME>` location is visible across multiple computers, Shared Services can be installed to only one computer. If you try to install Shared Services to another computer, the previous installation is detected.
- The wizard installs Shared Services files and shared components, such as JRE files.

  **Note:** For AIX, the Shared Services installer does not perform a full installation of JDK. See "Installation of JDK on AIX" on page 14.
- A typical installation installs these components:
    - OpenLDAP database (stores the security-services-related data)
    - Shared Services server
    - Shared Services documentation
- A custom installation installs only the components that you select.
- For information about resolving installation issues, see the *Hyperion Installation and Configuration Troubleshooting Guide*.
- You must install and configure a supported database. For a list of supported relational databases, see the *Hyperion Installation Start Here*.
- After installation is complete, the installation wizard prompts you to launch Hyperion Configuration Utility, which enables you to perform key configuration tasks. See "What's Next?" on page 21.

## Running Silent Installations

To install Shared Services on multiple computers and duplicate installation options, you can record your installation settings and run a silent installation from the command line. Silent installations automate the installation process so you can install Shared Services without repeating installation-setting specifications.

To record your installation settings, you create a response file and run the installation to record the settings in the response file. Then, when you run a silent installation, the response file is used to set the installation options of the nonsilent installation.

➤ To record installation settings and run a silent installation:

1 **Navigate to the directory that contains the product installer.**

2 **From a command line, run a command:**

| | |
|---|---|
| AIX, Solaris, HP-UX, and Linux | `setup.bin -options-record responsefilename`<br>For console mode:<br>`setup.bin -console -options-record responsefilename` |

**Note:** *responsefilename* can include an absolute path or file name.

The nonsilent product installer is launched.

3 **As you step through the installer, specify settings.**

Installation options are recorded in the response file. You can modify the response file to change installation options.

You are now ready to run the installation in silent mode.

4 **From the command line, enter a command:**

| | |
|---|---|
| AIX, Solaris, and HP-UX | `setup.bin -options <responsefilename> -silent` |
| Linux | `setup.sh -options <responsefilename> -silent` |

The installation runs in the background.

## Enabling Hyperion Remote Authentication Module

Oracle's Hyperion® Remote Authentication Module™ is an optional component for external authentication. If you are implementing security using an NTLM provider and using UNIX on the computer where the Hyperion application software is installed, ensure that Hyperion Remote Authentication Module is installed.

Hyperion Remote Authentication Module, formerly NTLM Remote Server, can be installed from Oracle's E-Delivery Web site. Installing and running Hyperion Remote Authentication Module can be useful to two groups:

- UNIX application users who must log on using a Windows NTLM domain
- Windows users who must log on using multiple Windows NTLM domains when no trust relationships are defined

**Note:** Earlier releases of Hyperion applications that employ Hyperion Remote Authentication Module 7.0 return an incorrect list of users and groups if the domain element is specified in the configuration file. To avoid this situation, use only Hyperion Remote Authentication Module 7.2.*x* or later with 7.2.*x* or later of Hyperion

Hub or Shared Services. For releases of Hyperion Hub prior to 7.2.*x*, continue to use Hyperion Remote Authentication Module 7.0.

For detailed information about Hyperion Remote Authentication Module, see the *Hyperion Security Administration Guide*.

## Installing Hyperion Remote Authentication Module

The Hyperion Remote Authentication Module installation program, `setup.exe`, is available on Oracle's E-Delivery Web site.

➤ To install Hyperion Remote Authentication Module:

1. Run **`setup.exe`**.
2. Select the language to use.
3. On the **Welcome** page, click **Next**.
4. Accept the license agreement, and click **Next**.
5. Choose a destination location for Hyperion Remote Authentication Module, and click **Next**.

   Hyperion recommends that you accept the default location.
6. Provide a value for the <*HYPERION_HOME*> environment variable.

   Hyperion recommends that you accept the default value.
7. Enter the host name and port number for the computer that hosts Hyperion Remote Authentication Module.

   The default port number is 58000.
8. If you are using Secure Sockets Layer with your NTLM deployment, select an option to support SSL.

   For SSL configuration, you must provide a value for the `<authProtocol></authProtocol>` element in the security-platform configuration XML file shipped with Shared Services (or alternately, the file shipped with an individual Hyperion product). See the *Hyperion Security Administration Guide*.

   **Note:** Hyperion Remote Authentication Module does not support SSL connections on AIX.
9. Click **Next**.
10. Review the summary of your installation choices, and click **Next**.
11. Click **Finish**.

## Configuring and Starting Hyperion Remote Authentication Module

➤ To configure and start Hyperion Remote Authentication Module:

1. On the computer that hosts the Hyperion products that connect to Hyperion Remote Authentication Module, modify the values in the `<location>` tags in the `<remoteServer>` section of the configuration file, to tell the application where to find Hyperion Remote Authentication Module.

You must provide a value for the `<remoteServer></remoteServer>` element in the external-authentication configuration XML file shipped with Shared Services (or alternately, the file shipped with each Hyperion product).

2 Selecting **Start > Programs > Hyperion > Foundation Services > Hyperion Remote Authentication Module > Run Authentication Server**.

# What's Next?

After installation is complete, the installation wizard prompts you to launch Hyperion Configuration Utility. Hyperion Configuration Utility guides you through a series of pages to perform these Shared Services configuration tasks:

- Configuring a relational database for Shared Services
- Configuring a mail server for Shared Services
- Deploying Shared Services on an application server

For information about launching and running Hyperion Configuration Utility, see Chapter 3, "Configuring Shared Services."

For information about manually deploying Shared Services on an application server, see one of the following appendices:

- Appendix A, "Manual Deployment to WebLogic Application Server"
- Appendix B, "Manual Deployment to WebSphere Application Server"
- Appendix C, "Manual Deployment to Oracle Application Server"

# 3 Configuring Shared Services

**In This Chapter**



## Hyperion Configuration Utility

Hyperion Configuration Utility is a common tool that installs automatically with Hyperion products. Although you must use it to set up new products that you install, it also enables you to reconfigure existing products and upgraded products. Configuration involves these tasks:

- Relational database configuration—To store and retrieve application data in a database repository.
- Mail server configuration—To use task automation notifications.
- Application server deployment—To deploy the application automatically, or partially, to an application server.

**Note:** You do not need to install Shared Services on every computer to which a Hyperion product is installed.

When using Hyperion Configuration Utility to configure Hyperion products other than Shared Services, you are given the option to register the products with Shared Services. You can use Shared Services only with products with which it is registered. See the installation guides of products that you want to register.

For information about the order of configuration tasks, acceptable characters, and resolving configuration issues, see:

- "Task Sequence" on page 24
- "Restricted Characters " on page 24
- "Troubleshooting " on page 24

## Task Sequence

Hyperion recommends that you configure Shared Services first, separately from other products and that you perform all Shared Services configuration tasks. Then use Hyperion Configuration Utility to configure other products installed on the computer.

Hyperion Configuration Utility performs Shared Services configuration tasks in this order:

- Database configuration
- Application server deployment
- Mail server configuration

All input is gathered by Hyperion Configuration Utility and configuration for these tasks occurs once at the end.

**Note:** Hyperion requires the database configuration and application server deployment tasks be executed the first time together; once done successfully, these tasks can be re–done separately.

## Restricted Characters

Only enter alphanumeric, dash (-), dot (.) , underscores (_), and tildes (~) during configuration. Tildes are only supported on Microsoft Windows. All other characters are not supported.

## Troubleshooting

Terminating configuration for one product does not stop the configuration of other products. All configuration warnings and errors are logged as follows:

`<HYPERION_HOME>/logs/config`

If you encounter errors, perform these tasks:

- Configure products individually.
- See the *Hyperion Installation and Configuration Troubleshooting Guide* for information about configuration checks, debugging using logs, troubleshooting methodology, and solutions to common configuration issues.

# Satisfying Initial Requirements

If you are using Hyperion Configuration Utility for the first time, perform these tasks:

**Table 3** Configuration Requirements

| Task | Reference |
|---|---|
| Satisfy system and product-specific requirements. | "System Requirements" and "Planning Hyperion Installations" in the *Hyperion Installation Start Here* |

| Task | Reference |
|---|---|
| Gather the information you need to configure products. | "Hyperion Configuration Utility Worksheets" in the *Hyperion Installation Start Here* |
| Install and configure Shared Services. | *Shared Services Installation Guide* |
| Install an application server | Application server documentation |
| Prepare a database to use for relational storage | Relational database documentation |

# Configuring Product Upgrades

Use Hyperion Configuration Utility to configure and reconfigure supported product upgrades. Note the following:

- If you upgraded Shared Services, configure it before configuring other products.
- Configure upgraded products individually.
- Deploy to the same database you used when you configured the previous product release.
- Hyperion Configuration Utility is backward compatible with previous Hyperion releases and can be used to configure Shared Services for previous releases.

# Configuring Shared Services

Run Hyperion Configuration Utility on the computer hosting the products to configure or reconfigure.

**Note:** Before beginning the configuration process, ensure that you have installed the application server you plan to use. Also, unlike other Hyperion products, you do not need to launch the Shared Services server to run Hyperion Configuration Utility for Shared Services.

➤ To configure Shared Services:

1 **Launch Hyperion Configuration Utility as follows:**

- At the end of installation by selecting **Launch Hyperion Configuration Utility** on the last panel.
- Using a method:

  On UNIX:

  - In XWindows, change to `<HYPERION_HOME>/common/config` and type `configtool.sh`.
  - In a console, change to `<HYPERION_HOME>/common/config` and type `configtool.sh – console`.

2 Select the language in which to configure and click **Next**.

3 On the Welcome page, click **Next**.

4 Select the products and the tasks to perform, then click **Next**.

5 Based on your selection, perform the following tasks, clicking **Next** between tasks.

**Table 4** Configuration Tasks

| Selection | Task |
|---|---|
| Configure Database | a. Start the database.<br>b. Select the database type.<br>c. Enter the information in "Configuring Databases" on page 27. |
| Deploy to Application Server | a. Start the application server.<br>b. Select the application server, then an option:<br>● Automatic– Hyperion Configuration Utility deploys all files to the application server, resulting in no or minimal post-deployment tasks:<br>❍ WebLogic: If disk space is inadequate, specify another location for the `WAR` file and redeploy.<br>**Note:** On WebLogic, a default username and password of hyperion is used internally for deployment.<br>❍ WebSphere: If disk space is inadequate, Hyperion Configuration Utility places `java.io.tempdir` in `<HYPERION_HOME>/temp`. After deployment, the `temp` folder is deleted.<br>● Manual– The `EAR` or `WAR` file is placed in this directory, enabling you to manually deploy after configuration:<br>`HSS_HOME>/<AppServer>/InstallableApps/common`<br>WebLogic 8.1.x – `<HSS_HOME>/<AppServer>/InstallableApps`<br>● Generic–This option facilitates manual deployment for application servers for which auto-deployment is not supported, for example, Oracle Application Server. The `EAR` or `WAR` file is placed in this directory, enabling you to manually deploy after configuration:<br>`<HYPERION_HOME>/deployments/generic`<br>After configuration, perform the deployment tasks for your application server in one of these appendices:<br>❍ Appendix A, "Manual Deployment to WebLogic Application Server"<br>❍ Appendix B, "Manual Deployment to WebSphere Application Server"<br>❍ Appendix C, "Manual Deployment to Oracle Application Server"<br>c. Enter the information in "Deploying to the Application Server " on page 28. |
| Configure Mail Server | Configure the mail server if you plan to use the Shared Services taskflow management system.<br>Enter the name of the SMTP mail server.<br>For example, `sendmail` might be the SMTP server for e-mail Other common mail server programs are `qmail` and `Exim`. |

6 Click **Finish**.

Configuration time depends on the products and tasks you selected. Progress is recorded in `configtool.log` as follows:

<HYPERION_HOME>/logs/config

When configuration finishes, the status of each task is displayed.

If configuration is successful, perform any required post-configuration tasks and start the product.

If errors display, perform these tasks:

- Configure products individually and perform tasks separately.
- See the *Hyperion Installation and Configuration Troubleshooting Guide* for information about resolving configuration issues.

## Configuring Databases

When you configure Shared Services to use a database, Hyperion Configuration Utility ensures that the database is connected and is a supported database type. If a database is detected, you may be prompted to choose whether to use the detected repository or create a repository. For Shared Services, this affects both the relational and the OpenLDAP databases.

If errors occur during a multiple-product configuration, terminating configuration for a particular product does not terminate the entire process. Configuration continues for the other products. Hyperion Configuration Utility displays error messages on a summary page after the configuration process completes.

For a list of supported databases for this release, see the *Hyperion Installation Start Here*. For database prerequisites, see "Satisfying Initial Requirements" on page 24.

**Table 5** Database Configuration

| Field | Instruction |
|---|---|
| Server | Name of the computer or server hosting the database. |
| Port | Server port number on which the database listens: |
| Product | Name of each product and its installation location. |
| Database or SID (Oracle only) | Database name or the Oracle system identification (database instance). Do not use restricted characters. If you want to accept the default, you must have created a database and database user using the default names. |
| Username | The name of the database owner. If you want to accept the default, you must have created a database and database user using the default names. |
| Password | The password of the database owner.<br>**Note:** If this changes, reconfigure as described in the *Hyperion Installation Start Here*. |
| Data Tablespace (Oracle) | Name of an existing tablespace used to create tables. The data tablespace is the logical portion of the database used to allocate storage for table data. |
| Index Tablespace (Oracle) | Name of an existing tablespace used to create database indexes. The index tablespace is the logical portion of the database used to allocate storage for index data. |

## Deploying to the Application Server

To prevent the Web application being deployed from inheriting unwanted runtime settings, you must create and use one of the following:

- Oracle OC4J instance
- WebLogic server
- WebSphere application server

Deploying more than one Web application to the same OC4J instance, WebLogic server, or WebSphere application server may yield unsuccessful results.

**Note:** If deploying Shared Services (automatically or manually) to WebLogic 9.2, the `<enforce-valid-basic-auth-credentials>` tag must be set to `false` in the last line of the `<security-configuration>` section in the WebLogic `config.xml` file (for the users domain, usually in `/bea/user_projects/domains/mydomain/config/config.xml`). This prevents WebLogic from trying to authenticate basic authentication headers.

**Caution!** The user account that installs and configures the Hyperion product being deployed must have permission to create a WebSphere profile. Refer to the IBM InfoCenter for detailed instructions on granting permission to create a WebSphere profile as a non-root user.

If you are deploying your product to a single application server, decide between these deployment options:

- Automatic—Select the Deploy to Application Server task and the Automatic deployment type in Hyperion Configuration Utility to have Hyperion Configuration Utility deploy all files to the application server. In most cases, no other deployment tasks are required.
- Manual—Select the Deploy to Application Server task and the Manual deployment type in Hyperion Configuration Utility to have Hyperion Configuration Utility place the necessary Web archives (EAR or WAR) in the `<`*`HSS_HOME`*`>/AppServer/InstallableApps/common` directory to enable manual deployment at a future time.

  **Note:** For WebLogic 8.1.*x*, the .EAR or .WAR file is placed in `<`*`HSS_HOME`*`>/AppServer/InstallableApps`.

- Generic—This option facilitates manual deployment for application servers for which auto-deployment is not supported, such as Oracle Application Server. Select the Deploy to Application Server task and the Generic deployment type in Hyperion Configuration Utility to have Hyperion Configuration Utility place the necessary Web archives (EAR or WAR) in the `<`*`HYPERION_HOME`*`>/deployments/generic` folder.

After configuration, perform the deployment tasks for your application server in one of these appendices:

- Appendix A, "Manual Deployment to WebLogic Application Server"
- Appendix B, "Manual Deployment to WebSphere Application Server"
- Appendix C, "Manual Deployment to Oracle Application Server"

**Note:** If you are redeploying the application server for Shared Services, you must also reconfigure the relational database.

**Table 6** Auto-Deployment Parameters

| Field | Description |
|---|---|
| Location | Path to the application server installation directory, for example:<br>• WebSphere Base:<br>`/opt/WebSphere/AppServer`<br>• WebSphere Express:<br>`/opt/IBM/WebSphere/Express51/AppServer`<br>• WebLogic 8.1.*x*:<br>`/opt/bea/weblogic81`<br>• WebLogic 9.1.*x*:<br>`/opt/bea/weblogic91` |
| Profile (WebSphere) | Name of the profile created by Hyperion Configuration Utility. By default, all applications deploy to the same profile. To change the profile name, see "What Happens During Deployment " on page 29. |
| Domain (WebLogic) | Default name of the domain where you access the application. For WebLogic 9.1.*x*, all applications deploy to the same domain. To change the domain name, see "What Happens During Deployment " on page 29. |
| Component | Products being deployed. Some products display as components. |
| Server Name | Enter the name of the server where you will access the product. Do not include spaces. This name is used as the product directory name in `<HYPERION_HOME>/deployments`. |
| Port | To change the default port, enter a unique port number that does not exceed 1025 to avoid conflicts with third-party port assignments. See "Ports" in the *Hyperion Installation Start Here*. |

# What Happens During Deployment

## WebSphere and WebLogic 9.1.*x*

Hyperion Configuration Utility deploys each application to the same WebSphere profile or WebLogic domain. The profile or domain is created when the first application is deployed. Each application runs in a separate JVM.

Hyperion Configuration Utility deploys the application to:

`<HYPERION_HOME>/deployments/<AppServNameAndVersion>`

Under this directory, the `bin` directory contains start and stop scripts for all deployed applications. For each application there is also a `setCustomParams<Product>.bat` file or a shell script where JAVA_OPTIONS can be changed when starting using start scripts.

To change the default profile or domain directory, modify the deployment directory parameter in the `weblogic.properties` or `websphere.properties` in:

`<HYPERION_HOME>/common/config/resources/<AppServName>/resources`

**Note:** It is not recommended to change other parameters in this file.

### WebLogic 8.1.*x*

Deploying to a single domain for WebLogic 8.1.*x* is not supported. For WebLogic 8.1.*x*, Hyperion Configuration Utility deploys the application to:

`HSS_HOME/AppServer/InstalledApps/<AppServName>/<Version>`

# Postconfiguration Tasks

## Backing Up Shared Services Configuration Files

Hyperion recommends that you back up Shared Services configuration files. If configuration files are not backed up, upon recovery, you must reconfigure Shared Services.

See Appendix D, "Shared Services Backup and Recovery."

## Starting and Stopping Shared Services

### Starting Shared Services

➤ To start Shared Services server manually, execute the startup script:

| Application Server | Path to Script |
|---|---|
| IBM WebSphere | `<HYPERION_HOME>/deployments/<AppServNameAndVersion>/bin/startSharedServices9.sh` |
| BEA WebLogic 8.1.*x* | `<HSS_HOME>/AppServer/InstalledApps/<AppServName>/<version>/SharedServices9/bin/startSharedServices9.sh` |

| Application Server | Path to Script |
|---|---|
| BEA WebLogic 9.1.*x* | <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/bin/startSharedServices9.sh |
| Oracle | To start Oracle Enterprise Manager:<br><*OracleInstallDir*>/bin/emctl start iasconsole<br>To start all managed applications under Oracle Enterprise Manager:<br><*OracleInstallDir*>/opmn/bin/opmnctl startall<br>To start OC4J instance:<br><*OracleInstallDir*>/opmn/bin/opmnctl start process-type=<*instance-name*><br>where Shared Services has been deployed to instance "<*instance-name*>". |
| Apache Tomcat | <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/bin/startSharedServices9.sh |

## Verifying Successful Startup of Shared Services

➤ To verify successful startup and configuration of Shared Services:

1 **During startup, look for the following confirmation messages in the Shared Services console window:**

- `Database Configuration Test Passed`
- `Security System Initialized Successfully`

  **Note:** This message will not display for Tomcat.

- `Shared Services Initialized Successfully`

  When Shared Services is deployed to the Tomcat application server, confirmation messages are logged to <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/SharedServices9/logs/Catalina.out.

  When Shared Services is deployed to WebSphere, the confirmation message is logged to <*WebSphereInstallDir*>/AppServer/logs/SharedServices9/SystemOut.log.

  When Shared Services is deployed to WebLogic 8.1.*x*, if the log level is not set to WARN, the confirmation message is logged to <*HSS_HOME*>/AppServer/InstalledApps/WebLogic/8.1/SharedServices9Domain/logs/SharedServices_Metadata.log.

  When Shared Services is deployed to WebLogic 9.1.*x*, if the log level is not set to WARN, the confirmation message is logged to <*HYPERION_HOME*>/deployments/WebLogic9/SharedServices9/logs/SharedServices_Metadata.log.

2 **On the Shared Services server computer, launch the User Management Console login page by opening a browser and entering this URL:**

`http://SharedServicesServerName:port#/interop`

where *SharedServicesServerName* is the name of the computer where the Shared Services server is installed and *port#* is the port number of the Shared Services server. The default port number is 58080; if Shared Services server is installed to a non-default port, specify that value. For example, using the default port:

```
http://jdoe:58080/interop/
```

**Note:** As a best practice, the URL should use an IP address or a fully qualified machine name that includes the domain name. If the IP address is dynamic, use the fully qualified machine name.

Display of the User Management Console login page indicates that the Shared Services server started successfully.

## Stopping Shared Services

➤ To stop Shared Services server manually:

1 **Execute the stop script:**

| Application Server | Path to Script |
|---|---|
| IBM WebSphere | <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/bin/stopSharedServices9.sh |
| BEA WebLogic 8.1.*x* | <*HSS_HOME*>/AppServer/InstalledApps/<*AppServName*>/<*version*>/SharedServices9/stopSharedServices9.sh |
| BEA WebLogic 9.1.*x* | <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/bin/stopSharedServices9.sh |
| Oracle | To stop Oracle Enterprise Manager:<br><*OracleInstallDir*>/bin/emctl stop iasconsole<br>To stop all managed applications under Oracle Enterprise Manager:<br><*OracleInstallDir*>/opmn/bin/opmnctl stopall<br>To start OC4J instance:<br><*OracleInstallDir*>/opmn/bin/opmnctl stop process-type=<*instance-name*><br>where Shared Services has been deployed to instance "<*instance-name*>". |
| Apache Tomcat | <*HYPERION_HOME*>/deployments/<*AppServNameAndVersion*>/bin/stopSharedServices9.sh |

2 **On WebLogic, if a message that suggests using the FORCESHUTDOWN command is displayed, use the FORCESHUTDOWN command to stop Shared Services server:**

a. In a text editor, open the stop script.

See the table under Step 1 for the stop script location.

b. In the file, find SHUTDOWN, and replace it with FORCESHUTDOWN.

c. Save and execute the file.

# WebLogic Postconfiguration Tasks

## Patch for WebLogic 8.1.6

There is a patch from BEA for WebLogic 8.1.6 to make it work with Shared Services. Without this patch, basic authentication fails.

Before you begin, install Shared Services and deploy to WebLogic 8.1.6 and any database. For instructions, see Chapter 2, "Installing Shared Services," Chapter 3, "Configuring Shared Services," and, if manually deploying, Appendix A, "Manual Deployment to WebLogic Application Server."

➤ To download and install the patch:

1. **Go to the BEA Support site (http://support.bea.com), open a case, and ask for Patch ID "CR287255" for WebLogic 8.1.6.**
2. **After you receive the download file from BEA Support, unzip it to any location (for example, `<BEA_HOME>/WLPatch`).**
3. **Edit `SET PRE_CLASSPATH` in the `startSharedServices9.sh` file located in:**

   `<HSS_HOME>/AppServer/InstalledApps/WebLogic/8.1/SharedServices9Domain/bin`
4. **Place the first argument for `SET PRE_CLASSPATH` as the location of the patch. For example, `"SET PRE_CLASSPATH=<BEA_HOME>/WLPatch/CR287255_810sp6.jar`.**
5. **Edit `config.xml` located in `<HSS_HOME>/AppServer/InstalledApps/WebLogic/8.1/SharedServices9Domain`:**

   Add `EnforceValidBasicAuthCredentials="false"` to the `SecurityConfiguration` tag

   ```
   <SecurityConfiguration

   CredentialEncrypted="{3DES}n/ZYsNUiTbdBCYmpc4rasOlvDr0/
   72Ub7CVpCjn62jtkIFbB624SwCSVCNgN/iTaa9a57dplMY0hx4F+DE0G6+YS8dox5x5E"

   EnforceValidBasicAuthCredentials="false"

   Name="SharedServices9Domain" RealmBootStrapVersion="1"/>
   ```
6. **Start Shared Services server.**

## Deploying WebLogic When Connected Through a Proxy Server

The process of deploying a product to WebLogic through a proxy server is automated; no manual steps are required.

## Enabling HTTPS for WebLogic

➤ To enable HTTPS support on WebLogic:

1 **Deploy Shared Services on WebLogic by using Hyperion Configuration Utility.**

For instructions see, "Deploying to the Application Server " on page 28.

2 **Enable SSL on WebLogic 8.1.*x*:**

a. Start Shared Services server on WebLogic:

`<HSS_HOME>/AppServer/InstalledApps/<AppServName>/<version>/SharedServices9/bin/startSharedServices9.sh`

b. Log on WebLogic Administration Console:

i. Access `http://hostname:58080/console`.

ii. Specify the username and password:

- Username: hyperion
- Password: hyperion

c. Click **Servers**.

d. Click **SharedServices9**.

e. Under **Change Center**, click **Lock Edit**.

f. Under the **General** tab, enable the **SSL Port Enabled** option.

g. Click **Activate**.

h. Log off WebLogic Administration Console.

3 **Enable SSL on WebLogic 9.1.*x*:**

a. Start Shared Services server on WebLogic:

`<HYPERION_HOME>/deployments/<AppServNameAndVersion>/bin/startSharedServices9.sh`

b. Log on WebLogic Administration Console:

i. Access `http://hostname:7001/console`.

ii. Specify the username and password:

- Username: hyperion
- Password: hyperion

c. Click **Servers**.

d. Click **SharedServices9**.

e. Under **Change Center**, click **Lock Edit**.

f. Under the **General** tab, enable the **SSL Port Enabled** option.

g. Click **Activate**.

h. Log off.

4 **Restart Shared Services server on WebLogic.**

HTTPS is enabled on port 7002. You can access Shared Services by using the URL `https://servername:7002/interop`.

### Updating the Default Session Time-out for WebLogic 8.1 on HP-UX

Because the default session time-out for WebLogic applications is 3600 seconds (1 hour), for HP-UX, Shared Services deployments to WebLogic 8.1 servers can require over 30 minutes to shut down the application server.

➤ To reduce the session-timeout setting:

1 **Locate the** `weblogic.xml` **configuration file, in** `<HSS_HOME>/AppServer/ InstalledApps/WebLogic/8.1/SharedServices9/interop/WEB-INF`.

2 **To the** `<session-param>` **section of** `weblogic.xml`, **add:**

```
<param-name>TimeoutSecs</param-name>

<param-value>timeout value in seconds</param-value>
```

For example:

```
<session-param>

      <param-name>TimeoutSecs</param-name>

      <param-value>120</param-value>

</session-param>
```

## Reconfiguring Products

Hyperion Configuration Utility enables you to reconfigure products to incorporate changes in your environment such as a different application server.

To reconfigure, launch Hyperion Configuration Utility on the computer hosting the product, and follow the procedures in this chapter.

**Note:** If you reconfigure a database, restart the application server afterward.

## What's Next?

After configuration is complete, the next task you must complete is to enable external authentication. For information about enabling external authentication, see the *Hyperion Security Administration Guide*.

**Note:** For Netegrity SiteMinder, install Oracle's Hyperion® Remote Authentication Module™ and configure Shared Services to use Single Socket Layers. For more information, see "Enabling Hyperion Remote Authentication Module" on page 19.

# 4 Uninstalling Shared Services

Shared Services provides a cross-platform uninstaller program that helps you remove Shared Services or Hyperion Hub from your system.

The uninstaller does not remove common components residing in the <*HYPERION_HOME*>/common directory, and does not uninstall the database that contains data.

**Note:** You do not need to uninstall previous releases before installing this release. If you are upgrading Shared Services or Hyperion Hub, review the upgrade procedures in Chapter 5, "Upgrading Shared Services" because you may need to manually back up files before uninstalling.

➤ To uninstall Shared Services and Hyperion Hub:

1 **Stop all activities and processes connected to Shared Services or Hyperion Hub: application server, OpenLDAP database, and servers related to Shared Services or Hyperion Hub, except IBM WebSphere (WebSphere should be left running because it removes files during uninstallation).**

**Note:** You do not need to stop the relational database.

To stop Shared Services, see "Stopping Shared Services" on page 32.

2 **Launch the uninstaller:**

**Table 7** Shared Services and Hyperion Hub Uninstaller Location

| Release | Uninstaller Location |
|---|---|
| Hyperion Hub Release 7.0 | /home/*username*/Hyperion/hyperionhub/7.0/uninstall/uninstallhyperionhub7.0.bin |
| Hyperion Hub Release 7.0.1 | /home/*username*/Hyperion/hyperionhub/7.0.1/uninstall/uninstallhyperionhub7.0.bin |
| Hyperion Hub Release 7.2.x | /home/*username*/Hyperion/hyperionhub/7.2.x/uninstall/uninstallhyperionhub7.2.bin |
| Shared Services Release 9.*x* | <*HSS_HOME*>/uninstall/uninstallHyperionSystemSharedServices9.bin<br>**Note:** <*HSS_HOME*> is the directory where Shared Services is installed. |

3 **Follow the uninstallation prompts.**

# 5 Upgrading Shared Services

In This Chapter



If you are upgrading from a previous release of Shared Services, note the following supported upgrade paths:

- 9.2.0.3 to 9.3.1
- 9.3.0.*x* to 9.3.1

If you are using a release prior to 9.2.0.3, you must first upgrade to one of the versions noted above, and then upgrade to 9.3.1. Upgrading from 9.2 releases prior to 9.2.0.3 will be supported in an upcoming service pack.

## Upgrade Considerations for User Directories

Shared Services 9.3.1 supports a unique identity attribute to locate users and groups in external LDAP-enabled user directories without reference to the location of their accounts. Implementation of this unique identity attribute enables Shared Services to maintain provisioning information even if users and groups are moved across organizational units (OUs) in external LDAP-enabled user directories.

While upgrading to Shared Services 9.3.1, you have the following options:

- Implement the unique identity attribute by reconfiguring your external user directory configuration.
- Keep the non-unique identity attribute (Distinguished Name), if Native Directory does not contain stale data (caused by user and group move across OUs, multiple users or groups with the same common name, or changes to the common name of provisioned users).

See the following sections in the *Hyperion Security Administration Guide* for detailed information:

- Handling User and Group Move Across OUs in LDAP-Enabled User Directories
- Configuring Oracle Internet Directory, MSAD, and Other LDAP-Enabled User Directories

**Caution!** If Shared Services implements the unique identity attribute, each Hyperion product must perform product-specific steps. After you migrate Shared Services users and groups to the unique identity attribute, Hyperion products stop working until the user and group information contained in product-specific repositories is updated to reflect the unique identity attribute. For product-specific instructions, see the "Upgrading" chapter in the product installation guide.

# Upgrading From Release 9.2.0.3 or 9.3.*x*

Follow this procedure to upgrade from release 9.2.0.3 or 9.3.*x* of Shared Services. You can install this release of Shared Services without uninstalling the previous release.

➤ To upgrade from release 9.2.0.3 or 9.3.*x* of Shared Services:

1 **Verify that all preparation tasks are complete and all system requirements are met.**

See *Hyperion Installation Start Here*.

2 **Stop all activities and processes connected to Shared Services, including the application server, the OpenLDAP database, and all servers related to Shared Services.**

**Note:** You do not need to stop the relational database.

To stop Shared Services manually, execute the stop script:

**Table 8** Shared Services Stop Scripts

| Application Server | Path to Script |
|---|---|
| IBM WebSphere | `<HSS_HOME>/AppServer/InstalledApps/<AppServName>/<version>/SharedServices9/bin/stopSharedServices9.sh` |
| BEA WebLogic | `<HSS_HOME>/AppServer/InstalledApps/<AppServName>/<version>/SharedServices9/stopSharedServices.sh` |
| Oracle | `<OracleInstallDir>/bin/emctl stop iasconsole`<br>`<OracleInstallDir>/opmn/bin/opmnctl stopall` |
| Apache Tomcat | `<HSS_HOME>/AppServer/InstalledApps/<AppServName>/<version>/SharedServices9/bin/stopSharedServices9.sh` |

3 **If the previous installation of Shared Services was a manual deployment, search on your system for the external authentication configuration file (`CSS.xml`), and manually back it up.**

4 **Install this release of Shared Services over the existing installation.**

See "Launching Installers" on page 17 and "Running the Installation Wizard" on page 17. Do not start Shared Services.

**Note:** You must install the Shared Services files to the previous-installation directory.

5 **Using Hyperion Configuration Utility, configure the Shared Services application server, RDBMS, and mail server.**

See Chapter 3, "Configuring Shared Services."

**Note:** You must use the existing database.

6 **Start Shared Services.**

**Note:** You must ensure Shared Services is started before configuring other Hyperion products.

If you upgrade Shared Services but do not upgrade other Hyperion products, you must re-register the other products with Shared Services using Hyperion Configuration Utility. For instructions, see the product installation guide.

# A Manual Deployment to WebLogic Application Server

**In This Appendix**



This appendix provides information about manually deploying Shared Services to WebLogic application server.

**Note:** These instructions are specific to the Shared Services 9.3.1.0.03 Service Fix. For information about the issues addressed in the Service Fix, see the Shared Services 9.3.1.0.03 Readme.

## Deployment Options

You can deploy automatically or manually:

- Automatic deployment—Use Hyperion Configuration Utility to automatically deploy all files to WebLogic application server. In most cases, no other deployment tasks are required. Select the Automatic deployment option for WebLogic application server. See Chapter 3, "Configuring Shared Services."
- Manual deployment—Use Hyperion Configuration Utility to populate and place the product archive in the appropriate directory. Select the Manual deployment option for WebLogic application server. After configuration, perform the remainder of the deployment tasks for WebLogic as documented in this appendix.

## Location References

The procedures in this section use the following location references to refer to the directories on your system:

- *<BEA_HOME>* is the installation directory of the BEA WebLogic server; for example, `/opt/bea`.

- <*WLS_HOME*> is the WebLogic server directory; for example, `opt/bea/weblogic91`
- <*HSS_HOME*> is the directory in which you installed Shared Services.
- <*HYPERION_HOME*> is the directory you specified during the Hyperion product installation.
- <*HSS_WL_HOME*> is the location where Hyperion Configuration Utility puts the files needed to manually deploy Shared Services:

  For WebLogic 8.1.6: <*HSS_HOME*>`/AppServer/InstalledApps/WebLogic/8.1/SharedServices9Domain`

  For WebLogic 9.1: <*HYPERION_HOME*>`/deployments/WebLogic9`

# Prerequisites

Complete these tasks before beginning:

- Install WebLogic application server and ensure it is running (see the application server documentation).
- Install Shared Services to the same computer as the application server (see Chapter 2, "Installing Shared Services").

  **Note:** Shared Services cannot be installed to directories with names containing spaces; for example, `$HOME/Program Files`. In addition, the system PATH environment variable should not contain spaces.

- After installation, run Hyperion Configuration Utility and configure Shared Services. If you are performing a manual deployment, select the "Deploy to Application Server" and "Configure Database" tasks. Provide the database information, and then select the manual deployment option on the Application Server selection panel (see Chapter 3, "Configuring Shared Services"). Selecting the manual deployment option copies the necessary files to:
  - For WebLogic 8.1.*x*, <*HSS_HOME*>`/AppServer/InstalledApps/WebLogic/8.1`
  - For WebLogic 9.1.*x*, <*HYPERION_HOME*>`/deployments/WebLogic9`
- If deploying Shared Services to WebLogic 9.2, the `<enforce-valid-basic-auth-credentials>` tag must be set to `false` in the last line of the `<security-configuration>` section in the WebLogic `config.xml` file (for the users domain, usually in `/bea/user_projects/domains/mydomain/config/config.xml`). This prevents WebLogic from trying to authenticate basic authentication headers.
- Start OpenLDAP (Native Directory) by running <*HYPERION_HOME*>`/SharedServices/`<*version*>`/openLDAP/startOpenLDAP.sh`.

# Deployment Instructions

This section provides detailed instructions for deploying to WebLogic application server. You can manually configure BEA WebLogic versions 8.1.6 and 9.1.

**Note:** To prevent the Web application being deployed from inheriting unwanted runtime settings, you must create and use a new WebLogic server. Deploying more than one Web application to the same WebLogic server instance may yield unsuccessful results.

See "Location References" on page 43 for the list of location references used in this section to refer to the directories on your system.

Before you begin deploying, review the "Prerequisites" on page 44.

➤ To configure the WebLogic server for Shared Services:

1 **Start the NodeManager service by running the `startNodeManager.sh` script located in the `<WLS_HOME>/server/bin` directory.**

2 **Create a new WebLogic domain or use an existing domain.**

Please refer to the BEA WebLogic documentation for instructions to create domains. Shared Services does not have any special requirements for creating domains.

3 **Start WebLogic by running the `startWebLogic.sh` file located in the domain directory.**

4 **Log on to WebLogic Administration Console:**

a. Open a browser and set the address to `http://server:portno/console` where *portno* is the listen port (specified during installation) on which the WebLogic server is running.

b. Sign in using the username and password you specified when creating a domain.

5 **Create a new server:**

a. Select the option to create/configure a new server. Enter the server name as `SharedServices` and the server listen port as `58080`.

**Note:** The NodeManager service must be running and a machine must be configured.

b. Select **Servers** and click **SharedServices**. Select the machine name and save the changes.

c. In the server's **Advanced Properties**, specify the following in the **Prepend to Classpath** field (separated by semi-colons — do not use carriage returns):

`<HYPERION_HOME>/common/JDBC/DataDirect/3.6/lib/hyjdbc.jar;`

`<HSS_WL_HOME>/SharedServices9/config;`

`<HSS_WL_HOME>/SharedServices9/config/lib/HubProductBean.jar;`

**Note:** See "Location References" on page 43 for the location of `<HSS_WL_HOME>`.

d. In the server's **Server Start** tab, specifiy the following in the **Arguments** field:

`-Dhyperion.home=<HYPERION_HOME>`

where `<HYPERION_HOME>` is the directory where the Hyperion common files were installed.

e. Add the following to the Native Library Path in <*WL_DOMAIN*>/startWeblogic.sh, where Native Library Path is an environment variable appropriate for your operating system. For example, PATH on Windows, LD_LIBRARY_PATH on UNIX/Linux, LIBPATH on AIX, or SHLIB_PATH on HP-UX:

<*HYPERION_HOME*>/common/SAP/bin (only if using SAP as a user directory)

6 **Update the Shared Services deployment location (<*HSS_WL_HOME*>):**

**Note:** See "Location References" on page 43 for the location of <*HSS_WL_HOME*>.

Copy the following files to the specified location:

- Copy the CSS.xml file from:

  <*HSS_WL_HOME*>/SharedServices9/config

  to:

  <*HSS_WL_HOME*>/servers/SharedServices9/webapps/interop/WEB-INF/classes

  **Note:** You must repeat this step each time a user directory is added or modified.

- Copy ALL *.properties files from:

  <*HSS_WL_HOME*>/SharedServices9/config

  to:

  <*HSS_WL_HOME*>/servers/SharedServices9/webapps/interop/WEB-INF/classes

- Copy xercesImpl.jar from:

  <*HYPERION_HOME*>/common/XML/JAXP/1.2.2

  to:

  <*HSS_WL_HOME*>/servers/SharedServices9/webapps/interop/WEB-INF/lib

- Copy jaxm-* from:

  <*HYPERION_HOME*>/common/XML/JAXM/1.1.1

  to:

  <*HSS_WL_HOME*>/servers/SharedServices9/webapps/interop/WEB-INF/lib

- Copy saaj-* from:

  <*HYPERION_HOME*>/common/XML/JAXM/1.1.1

  to:

  <*HSS_WL_HOME*>/servers/SharedServices9/webapps/interop/WEB-INF/lib

7 **Deploy the new Web application modules to the SharedServices server:**

a. From the WebLogic Administration Console, start the SharedServices server.

b. From the WebLogic Administration Console, select **Deployments** and deploy/install a new web application module.

c. Specify the path to the `interop` folder (`<HSS_WL_HOME>/servers/SharedServices9/webapps/interop`).

d. In the subsequent installation screens, select the following options:

- **Install this deployment as an application** (does not apply to WebLogic 8.1.*x*)
- The target server is **SharedServices**
- Retain the deployment name as **interop** (does not apply to WebLogic 8.1.*x*).
- Select **I will make the deployment accessible from the following location** and specify the full path to the `interop` folder.

e. Save and activate the changes.

f. Modify the `CSS.xml` file. See "Modifying the CSS.xml File" on page 47.

g. Stop and restart the `SharedServices` server.

h. For WebLogic 9.1 only: Select **Deployments** and then `interop` web application and click **Start -> servicing all requests**. Once the status of the application changes to "active", the application is available.

8 **Launch** `http://hostname:port/interop/`.

# Modifying the CSS.xml File

You need to modify the `CSS.xml` file as described in this section.

For WebLogic 8.1, this file is located in `<HSS_HOME>/AppServer/InstalledApps/WebLogic/8.1`. For WebLogic 9.1, this file is located in `<HYPERION_HOME>/deployments/WebLogic9/SharedServices9/config`.

In the `CSS.xml file`, ensure the hostname is used instead of `localhost`. The hostname is the name of the computer running OpenLDAP, not the computer running the Web server (if you have a Web server). Also ensure the correct port is being used for the OpenLDAP database.

For example, change the following values:

```
<hub location="http://localhost:58080">
   <dirPort>58089</dirPort>
</hub>
```

to these values:

```
<hub location="http://hostname:58080">
   <dirPort>58089</dirPort>
</hub>
```

# B Manual Deployment to WebSphere Application Server

**In This Appendix**



This appendix provides information about manually deploying Shared Services to WebSphere application server.

**Note:** These instructions are specific to the Shared Services 9.3.1.0.03 Service Fix. For information about the issues addressed in the Service Fix, see the Shared Services 9.3.1.0.03 Readme.

## Deployment Options

You can deploy automatically or manually:

- Automatic deployment—Use Hyperion Configuration Utility to automatically deploy all files to the application server. In most cases, no other deployment tasks are required. Select the Automatic deployment option for WebSphere application server. See Chapter 3, "Configuring Shared Services."
- Manual deployment—Use Hyperion Configuration Utility to populate and place the product archive in the appropriate directory. Select the Manual deployment option for WebSphere application server. After configuration, perform the remainder of the deployment tasks for WebSphere as documented in this appendix.

## Location References

The procedures in this section use the following location references to refer to the directories on your system:

- *<WAS_HOME>* is the installation directory of the WebSphere server; for example, `/opt/websphere`.

- *<HSS_HOME>* is the directory in which you installed Shared Services.
- *<HYPERION_HOME>* is the directory you specified during the Hyperion product installation.

# Prerequisites

Complete these tasks before beginning:

- Install WebSphere application server and ensure it is running (see the WebSphere documentation).
- In WebSphere, create a profile and a server.
- Install Shared Services to the same computer as the application server (see Chapter 2, "Installing Shared Services").

  **Note:** Shared Services cannot be installed to directories with names containing spaces; for example, `$HOME/Program Files`.

- After installation, run Hyperion Configuration Utility and configure Shared Services. If you are performing a manual deployment, select the "Deploy to Application Server" and "Configure Database" tasks. Provide the database information, and then select the manual deployment option on the Application Server selection panel (see Chapter 3, "Configuring Shared Services"). Selecting the manual deployment option enables database configuration to proceed automatically and creates the *<HYPERION_HOME>*`/deployments/WebSphere6` folder with files configured for the database. After configuration, perform the remainder of the deployment tasks for WebSphere as documented in this appendix.
- Start OpenLDAP (Native Directory) by running *<HYPERION_HOME>*`/SharedServices/`*<version>*`/openLDAP/startOpenLDAP.sh`.

**Note:** To prevent the Web application being deployed from inheriting unwanted runtime settings, you must create and use a new application server. Deploying more than one Web application to the same application server may yield unsuccessful results.

# Deployment Instructions

This section provides detailed instructions for deploying to WebSphere application server. See "Location References" on page 49 for the list of location references used in this section to refer to the directories on your system.

Before you begin deploying, review the "Prerequisites" on page 50.

➤ To manually deploy Shared Services to WebSphere application server:

1 **Start the WebSphere server.**

2 **Log on to the WebSphere Administration Console (as the administrator) using this URL:**

`http://`*servername*`:`*PortNumber*`/ibm/console`

For *servername* and *PortNumber*, enter the server and port on which WebSphere is running. The default location is `http://localhost:9060/ibm/console`.

3 **Create a server called `HSS`.**

4 **Create the `HYPERION_HOME` and `HSS_HOME` WebSphere environment variables:**

a. In the left frame, open the **Environment** folder, and click **WebSphere Variables**.

b. Click **New** and create the following variables:

- `HYPERION_HOME`: Set the value to the <*HYPERION_HOME*> directory you specified during installation. Set to server-level scope. The default is `/home/`*username*`/Hyperion`.
- `HSS_HOME`: Set the value to the <*HSS_HOME*> directory you specified during installation. The default is `/home/`*username*`/Hyperion/SharedServices/9.3.1`.

5 **Create a new shared library called Hyperion Shared Services Libraries:**

a. In the left frame, open the **Environment** folder, and click **Shared Libraries**.

b. Click **New**, name the shared library **Hyperion Shared Services Libraries**, and add the following libraries to the shared library in the classpath:

- Libraries:

```
${HYPERION_HOME}/deployments/WebSphere6/SharedServices9/config

${HYPERION_HOME}/common/JDBC/DataDirect/3.6/lib/hyjdbc.jar

${HYPERION_HOME}/common/JakartaCommons/commons-collections-3.2.jar

${HYPERION_HOME}/common/JakartaCommons/commons-dbcp-1.2.1.jar

${HYPERION_HOME}/common/JakartaCommons/commons-logging-1.1.jar

${HYPERION_HOME}/common/JakartaCommons/commons-pool-1.3.jar

${HYPERION_HOME}/common/JDBC/DataDirect/3.6/lib

${HYPERION_HOME}/common/SAP/lib

${HYPERION_HOME}/common/XML/JAXM/1.1.1/jaxm-api.jar

${HYPERION_HOME}/common/XML/JAXM/1.1.1/jaxm-runtime.jar

${HYPERION_HOME}/common/XML/JAXM/1.1.1/saaj-api.jar

${HYPERION_HOME}/common/XML/JAXM/1.1.1/saaj-ri.jar

${HYPERION_HOME}/deployments/WebSphere6/SharedServices9/config/
jdom-b9.jar
```

- And add the following path to the Native Library Path field:

```
${HYPERION_HOME}/common/CSS/9.3.1/bin
```

c. Click **OK**.

6 **Create or update virtual hosts:**

a. In the left frame, select **Environment** > **Virtual Hosts**.

b. Create or update a Virtual Host by defining a Host Alias for port 58080.

(By default, Shared Services is configured to run on port 58080.)

For details, please refer to the WebSphere documentation.

7 **Deploy the WAR file:**

a. In the left frame, select **Applications** > **Enterprise Applications**.

b. Click **Install**.

c. Navigate to `<HYPERION_HOME>/deployments/WebSphere6/SharedServices9/config/interop.war`.

d. Set the context root to `interop`, click **Next**.

**Caution!** Once you set the context root to `interop`, do not modify.

e. Click **Next** and accept the defaults.

f. Select the virtual host configured in the previous step and select **Hyperion Shared Services**, click **Next**, then click **Finish**.

g. After the **Application Server is Successfully Deployed** confirmation message is displayed, click **Save to Master Configuration**.

8 **Add a transport chain:**

a. In the WebSphere administrative console, click **Servers** > **Application servers** > **<server_name>**.

b. Under **Web container settings**, click **Web container transport chains**.

c. Click **New**.

The Create New Transport Chain wizard initializes. Specify the following information:

- A name for the new chain (for example, HSS).
- A transport chain template (for security).
- A port name (for example, HSS), the host name (you can enter *) or IP address for that port, and the port number under which Shared Services is going to run.

d. Click **Finish**.

e. Copy:

`<WAS_HOME>/profiles/<PROFILE_NAME>/installedApps/<CELL_NAME>/interop_war.ear/interop.war/WEB-INF/lib/jdom-b9.jar`

to:

`<HYPERION_HOME>/deployments/WebSphere6/SharedServices9/config`

9 **Add the Java argument** `-Dhyperion.home`**:**

a. Select **Application Servers** and then select the `HSS` server.

b. Select **Java and process management**.

c. Select **Process Definition** > **Java Virtual Machine**.

d. In the **Generic JVM Arguments** field, add `-Dhyperion.home=<HYPERION_HOME>`,

where `<HYPERION_HOME>` is the directory where the Hyperion common files were installed.

10 **Reference the shared library:**

For 6.1.0.5:

a. In the left frame, select **Applications** > **Enterprise Application**.

b. Select **`interop.war`**.

c. Under **Detail Properties**, select **Classloading and update detection**.

d. Under **General Properties**, for **Class loader order**, select **Class loaded with Application Class loader first**.

e. Set a reasonable value for the polling interval, and then click **OK**.

f. Under **References**, select **Shared Library References**.

g. Select the Hyperion Shared Services Module and select the **Hyperion Shared Services Libraries** Shared Library, and click **OK**.

For 6.0.2.11:

For classloading:

a. In the left frame, select **Applications** > **Enterprise Application**.

b. Select **`interop.war`**.

c. Under **General Properties**, select **Classloading and update detection**.

d. For **Class loader mode**, select **Parent Last**.

e. Set a reasonable value for the reloading interval, and then click **OK**.

To load shared libraries:

a. In the left frame, select **Applications** > **Enterprise Application**.

b. Select **`interop.war`**.

c. Under **Additional Properties**, select **Libraries** and add the Hyperion Shared Services Libraries created previously.

d. Save the configuration.

11 **Override session management:**

a. From the left frame, select **Enterprise Applications** and in the right frame, click **`interop.war`**.

b. In **Web Module Properties**, click **Session Management**.

c. Check **Override Session Management** to override the server setting.

d. Select **Enable Cookies**, change the cookie name to **HUBSESSIONID**, and click **Apply**.

12 **Select the Save command in the top menu bar to save the entire configuration.**

13 **Modify the Shared Services `CSS.xml` file. See "Modifying the CSS.xml File" on page 57.**

14 **Stop the WebSphere server application.**

WebSphere is now configured to run Shared Services.

15 **Start the WebSphere server application.**

16 **Log on Oracle's Hyperion® Shared Services User Management Console using this URL:**

`http://`*`ServerName`*`:`*`ServerPort`*`/interop`; for example, `http://localhost:58080/interop`.

## Increasing the Memory Allocation for JVM

If you share large models, you must increase the amount of memory WebSphere allocates for JVM.

➤ To increase the WebSphere memory allocation for JVM:

1 **Log on as the administrator to the WebSphere Administration Console by opening a browser and setting the address to `http://`*`servername`*`:`*`portno`*`/ibm/console` where *`servername`* and *`portno`* are the server and port (specified during installation) on which the WebSphere server is running.**

The default location is `http://localhost:9060/ibm/console`.

2 **In the left frame, select Servers > Application Servers; in the right frame, select SharedServices9 (or if you installed manually, select the server to which you installed Shared Services).**

3 **Select Additional Properties Process Definition.**

4 **Select Additional Properties Java Virtual Machine.**

5 **Set Initial Heap Size to 128. (Size is in megabytes.)**

6 **Set Maximum Heap Size to as much memory as you can allocate for the computer.**

Hyperion recommends a setting of 1024. (Size is in megabytes.)

7 **Click OK.**

8 **Click OK.**

9 **Select the Save menu command.**

10 **Restart the Shared Services server.**

## Troubleshooting the WebSphere Application Server Configuration

If you are unable to start Shared Services, follow the procedure in this section to troubleshoot the WebSphere server configuration.

➤ To troubleshoot the WebSphere server configuration:

1 **Start the WebSphere server.**

In most cases this is the WebSphere default server: `server1`. Type the following text at the command prompt: `./startServer.sh server1`

2 **Log on as the administrator to the WebSphere Administration Console using this URL:**

`http://servername:portno/ibm/console`

For *servername* and *portno*, enter the server and port on which WebSphere is running. The default location is `http://localhost:9060/ibm/console`.

3 **In the left frame, open the Servers folder, click Application Servers, and verify there is a server named SharedServices9.**

4 **In the left frame, open the Environment folder, click Manage Websphere Variables, and ensure the context is set to Cell=DefaultNode, Node=DefaultNode.**

5 **Ensure the <*HYPERION_HOME*> environment variable is set and that it points to the correct directory. If it does not exist, create it.**

6 **In the left frame, open the Environment folder, click Virtual Hosts, and verify there is a virtual host named** `hyperionVirtualHost`:

- If there is no virtual host named `hyperionVirtualHost`, follow these steps:

a. Click **New**.

b. Enter the name **hyperionVirtualHost** and click **OK**.

c. Select the **hyperionVirtualHost** link.

d. In **Additional Properties**, select the **Host Aliases** link. Click **Add**.

e. Enter a **Host Name** of *.

f. Enter the port number to use for Shared Services.

**Note:** Each application port number must be unique. If you modify a default port number, change it to a port number not currently used.

i. Click **OK** to create the Host Alias.

ii. Click **OK** to save the changes to `hyperionVirtualHost`

iii. Select **Save** to save the change.

- If there is a virtual host named `hyperionVirtualHost`:

a. Select the **hyperionVirtualHost** link.

b. In **Additional Properties**, select the **Host Aliases** link.

c. Select the link * and verify the port number is the one specified during Shared Services installation. This port number must not be used by other Web applications on your server. Take one of these actions:

d. If the port number is used by another application, correct the port number and click **OK** to save the change.

e. If the port number is correct, click **Cancel** to return to the previous screen.

f. Click **OK** to save the changes to `hyperionVirtualHost`.

g. If you changed the port number in step 6.c, select **Save Menu** to save the change.

7 **In the left frame, open the Environment folder, click Shared Libraries, and verify the scope is set as follows:**

Cell=DefaultNode, Node=DefaultNode, Server=SharedServices9

If the scope is not set correctly:

a. Expand the scope section.

b. Correct the information and click **Apply** to save the change.

8 **Verify there is a shared library at the server level called Hyperion Shared Services Libraries:**

- If the shared library does not exist, follow the procedure in the configuration section for creating this shared library.
- If the shared library exists, click the **Hyperion Shared Services Libraries** link and verify there is a carriage return after each class path. Click **OK**.
- If the shared library exists but was accidentally created at the node level, rename the server level library set and link it to the `interop.war` file (see step 10).

9 **In the left frame, open the Applications folder, click Enterprise Application, and verify there is an application called adminconsole_SharedServices9.**

If the application does not exist, you cannot use Hyperion Configuration Utility when the **SharedServices9** server is running. You can do what you are doing now, which is to start the `server1` server, make your changes, shut down `server1` and start the **SharedServices9** service.

10 **In the left frame, open the Applications folder, click Enterprise Application, and verify there is an application called interop.war:**

- If the **interop.war** application does not exist, create one:

a. Click **Install** and navigate to the file: `<HSS_HOME>/AppServer/InstallableApps/common/WebSphere/<version>/interop.war`.

b. Set the context root to `interop` and click **Next**.

> **Caution!** Once the context root is set to `interop`, do not modify.

c. Click **Next** to accept the defaults on the bindings page.

d. Click **Continue** on the Application Security Warning page.

e. Click **Next** to accept the defaults.

f. Select **hyperionVirtualHost** and select **Hyperion Shared Services**. Click **Next**.

g. Select **Hyperion Shared Services**.

h. Select the **Server WebSphere: cell=DefaultNode, node=DefaultNode**, and **server=SharedServices9.** Click **Apply** to move it to the Hyperion Shared Services Server.

i. Click **Next.** Click **Finish**.

j. Select **Save to Master Configuration** and click **Save**.

k. In the left frame, open the **Applications** folder and click **Enterprise Application**.

- If the **interop.war** application exists or if you just installed the **interop.war** application:

a. Select the **interop.war** link.

b. Select the **Additional Properties Libraries** link.

c. If the **Hyperion Shared Services Libraries** link is not displayed, click **Add**, select **Hyperion Shared Services Libraries**, and click **OK**.

d. Select **Additional Properties Session Management**.

e. Select **Enable Cookies** and ensure the cookie name is set to **HUBSESSIONID**. Click **OK** to return to the previous screen.

f. Ensure **Override** is selected. Click **OK** to return to the previous page.

g. Click **OK** to save your changes

h. Select the **Save** menu command to save your changes to the Master Configuration, and click **Save** again to save the changes.

i. In the `HubProductBean.jar` file, remove the classes related to the `jce1_2_2.jar` file.

   These are classes that are part of the package structure 'javax/crypto/' from `HubProductBean.jar`.

11 **Log off the Administration page and stop the application server.**

12 **Start the SharedServices9 Application Server by typing the following text at the command prompt:** `./startServer.sh server1`.

13 **In a Web browser, try to start the following Web page:**

```
http://localhost:9080/interop/framework/login
```

**Note:** Use the port number you specified during Shared Services installation.

If this page is not displayed, review the following log file: `<WAS_HOME>/WebSphere/AppServers/logs/server1/SystemOut.log`

You can ignore most of the errors generated because you have yet to set up HUB. However, if the log file says it cannot find the `Default.xml` file, HUB cannot start and you must check whether it is because of one of these reasons:

- The Hyperion Shared Services Libraries definition is incorrect and each class is not on its own line.
- The libraries are not defined for `interop.war`.
- The `<HYPERION_HOME>` environment variable is incorrect.

If after following these additional steps you still cannot start Shared Services, contact Hyperion Technical Support.

# Modifying the CSS.xml File

You need to modify the `CSS.xml` file as described in this section.

For WebSphere, this file is located in `<HYPERION_HOME>/deployments/WebSphere6/SharedServices9/config`.

In the `CSS.xml file`, ensure the hostname is used instead of `localhost`. The hostname is the name of the computer running OpenLDAP, not the computer running the Web server (if you have a Web server). Also ensure the correct port is being used for the OpenLDAP database.

For example, change the following values:

```
<hub location="http://localhost:58080">
   <dirPort>58089</dirPort>
</hub>
```

to these values:

```
<hub location="http://hostname:58080">
   <dirPort>58089</dirPort>
</hub>
```

# C Manual Deployment to Oracle Application Server

**In This Appendix**



## Overview

This appendix provides information about manually deploying Shared Services to Oracle application server.

In this chapter, <*HYPERION_HOME*> identifies the directory for Hyperion products. By default <*HYPERION_HOME*> is `/home/username/Hyperion`.

**Note:** These instructions are specific to the Shared Services 9.3.1.0.03 Service Fix. For information about the issues addressed in the Service Fix, see the Shared Services 9.3.1.0.03 Readme.

## Prerequisites

Complete these tasks before beginning:

- Install Oracle application server and ensure it is running (see the application server documentation).
- Install Shared Services to the same computer as the application server (see Chapter 2, "Installing Shared Services").

**Note:** Shared Services cannot be installed to directories with names containing spaces; for example, `$HOME/Program Files`.

**Note:** The operating system user who owns the Oracle Application Server (OAS) instance should be used to install Shared Services. If that is not possible, you must grant appropriate file-system access to the operating system user who owns the OAS instance. To operate properly, the OAS operating system user must be able to read all files in <*HYPERION_HOME*>, and must have write access to <*HYPERION_HOME*>/

deployments/generic/SharedServices9/config/CSS.xml and <*HYPERION_HOME*>/logs.

- After installation, run Hyperion Configuration Utility to configure Shared Services (see Chapter 3, "Configuring Shared Services").

  For Oracle application server, use Hyperion Configuration Utility to manually deploy Shared Services to the application server. Select the "Generic" application server type. Selecting "Generic" enables database configuration to proceed automatically and creates the <*HYPERION_HOME*>/deployments/generic folder with files configured for the database. After configuration, perform the remainder of the deployment tasks for Oracle as documented in this appendix.

- Start OpenLDAP (Native Directory) by running <*HYPERION_HOME*>/SharedServices/<*version*>/openLDAP/startOpenLDAP.sh.

# Deployment Instructions

This section provides detailed instructions for manually deploying to Oracle application server. You can manually configure Oracle versions 10.1.2.0.2 and 10.1.3.1.0.

Before you begin deploying, review the "Prerequisites" on page 59.

## Oracle 10.1.3.1

### Creating an OC4J Instance

To prevent the Web application being deployed from inheriting unwanted runtime settings, you must create and use a new OC4J instance. Deploying more than one Web application to the same OC4J instance may yield unsuccessful results.

Shared Services does not require modifications to Oracle HTTP Server listen port(s). However, if you wish to modify these ports, it is recommended that you do so before continuing. Refer to the appropriate Oracle® Application Server guide for details on changing port numbers.

➤ To create a new OC4J instance called SharedServices, follow the instructions in the *Oracle Containers for J2EE Configuration and Administration Guide*.

**Note:** OC4J group administrative settings such as environment variables are shared by all group members. The OC4J instance you create for the Web application being deployed should be assigned to a group created specifically for that application. (When an application is deployed to more than one OC4J instance, those instances can be assigned to the same group.) For example, when deploying Shared Services and Oracle's Hyperion® Workspace, the OC4J instance created for Shared Services should be assigned to a different group than the instance created for Workspace. Refer to the *Oracle® Process Manager and Notification Server Administrator's Guide* for more information about groups.

## Configuring the OC4J Instance

➤ To configure the SharedServices OC4J instance:

1 **In the OC4J instance home for the** SharedServices **instance, edit** config/global-web-application.xml **by adding the following block inside the** jsp <servlet> **definition:**

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

The complete block should resemble:

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>oracle.jsp.runtimev2.JspServlet</servlet-class>
  <!-- you can set main_mode to "justrun" to speed up
JSP dispatching, if you don't need to recompile
your JSP anymore. You can always switch your
main_mode. Please see our doc for details -->
  <!--
<init-param>
<param-name>main_mode</param-name>
<param-value>justrun</param-value>
</init-param>
-->
  <init-param>
    <param-name>req_time_introspection</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>
```

2 **Log on to Oracle Enterprise Manager 10g Application Server Control.**

3 **Start the** SharedServices **OC4J instance if it is not running.**

4 **From the** **Cluster Topology** **page in Application Server Control Console, click the** SharedServices **OC4J instance.**

5 **Click** **Administration** **and then** **Server Properties**.

6 **Under** **Command Line Options**, **increase the initial and maximum heap size; for example:**

```
Maximum heap size 512M

Initial heap size 256M
```

7 **Add the following Java option:**

```
-Dhyperion.home=<HYPERION_HOME>
```

where <HYPERION_HOME> is the directory where the Hyperion common files were installed.

8 **If Oracle Application Server is running on Windows, Linux, Solaris, or HP-UX, add the following Java options:**

```
-XX:PermSize=64m
-XX:MaxPermSize=128m
```

9 If Oracle Application Server is running on Solaris 10 for SPARC, add the following Java option as recommended by Metalink Note: 431167.1:

```
-Djava.nio.channels.spi.SelectorProvider=sun.nio.ch.PollSelectorProvider
```

10 Apply all changes, and then restart the `SharedServices` OC4J instance.

## Deploying interop.war

Use the Oracle Enterprise Manager 10g Application Server Control to deploy the `interop.war` Web archive to the `SharedServices` OC4J instance that you created.

➤ To deploy `interop.war`:

1 On the **Cluster Topography** page of the Oracle Enterprise Manager 10g Web console, select the `SharedServices` OC4J instance.

2 Click **Applications** and then **Deploy**.

3 Supply the absolute location to `interop.war` by selecting one of the following:

- Archive is present on the local machine...
- Archive is on the same machine...

For example: `<HYPERION_HOME>/deployments/generic/SharedServices9/config/interop.war`

4 Click **Next** to create a new deployment plan.

5 In **Application Name**, provide a name, for example, `SharedServices`.

6 Click **Next**.

7 In **Deployment Settings**, **Configure Class Loading**, add the following Java archives to the **Classpath** field, substituting `<HYPERION_HOME>` with the actual file-system location. Separate each entry with a semicolon (;).

```
<HYPERION_HOME>/deployments/generic/SharedServices9/config;

<HYPERION_HOME>/deployments/generic/SharedServices9/config/lib/
HubProductBean.jar;

<HYPERION_HOME>/common/JDBC/DataDirect/3.6/lib/hyjdbc.jar;

<HYPERION_HOME>/common/JakartaCommons/commons-collections-3.2.jar;

<HYPERION_HOME>/common/JakartaCommons/commons-dbcp-1.2.1.jar;

<HYPERION_HOME>/common/JakartaCommons/commons-logging-1.1.jar;

<HYPERION_HOME>/common/JakartaCommons/commons-pool-1.3.jar;

<HYPERION_HOME>/common/loggers/Log4j/1.2.14/lib/log4j-1.2.14.jar;

<HYPERION_HOME>/common/XML/JAXP/1.2.2/xercesImpl.jar

<HYPERION_HOME>/common/SAP/lib;
```

8 Click **Deploy**, and after successful deployment, restart the `SharedServices` OC4J instance.

### Post-Deployment Tasks

➤ Complete these tasks after deploying:

1 **Edit** `<HYPERION_HOME>/deployments/generic/SharedServices9/config/CSS.xml` **by modifying** `<hub location>` **with the correct hostname and port for your Oracle HTTP Server; for example:**

```
<hub location="http://hostname.example.com:8888">
```

2 **Restart the** `SharedServices` **OC4J instance.**

## Oracle 10.1.2.0.2

### Creating an OC4J Instance

To prevent the Web application being deployed from inheriting unwanted runtime settings, you must create and use a new OC4J instance. Deploying more than one Web application to the same OC4J instance may yield unsuccessful results.

Shared Services does not require modifications to Oracle HTTP Server listen port(s). However, if you wish to modify these ports, it is recommended that you do so before continuing. Refer to the appropriate Oracle® Application Server guide for details on changing port numbers.

➤ To create a new OC4J instance called `SharedServices`, follow the instructions in the *Oracle Containers for J2EE Configuration and Administration Guide*.

**Note:** OC4J group administrative settings such as environment variables are shared by all group members. The OC4J instance you create for the Web application being deployed should be assigned to a group created specifically for that application. (When an application is deployed to more than one OC4J instance, those instances can be assigned to the same group.) For example, when deploying Shared Services and Workspace, the OC4J instance created for Shared Services should be assigned to a different group than the instance created for Oracle's Hyperion® Workspace. Refer to the *Oracle® Process Manager and Notification Server Administrator's Guide* for more information about groups.

### Configuring the OC4J Instance

➤ To configure the `SharedServices` OC4J instance:

1 **Log on to Oracle Enterprise Manager 10g Application Server Control.**

2 **Start the** `SharedServices` **OC4J instance if it is not running.**

3 **On the Main page, in System Components, select the** `SharedServices` **OC4J instance.**

4 **Select Administration and then Server Properties.**

5 **Under Command Line Options, increase the initial and maximum heap size; for example:**

```
Maximum heap size 512M
```

```
Initial heap size 256M
```

6 **Add the following Java option:**

```
-Dhyperion.home=<HYPERION_HOME>
```

where <HYPERION_HOME> is the directory where the Hyperion common files were installed.

7 **If Oracle Application Server is running on Windows, Linux, Solaris, or HP-UX, add the following Java options:**

```
-XX:PermSize=64m
-XX:MaxPermSize=128m
```

8 **Apply all changes, and then restart the** SharedServices **OC4J instance.**

## Deploying interop.war

Use the Oracle Enterprise Manager 10g Application Server Control to deploy the interop.war Web archive to the SharedServices OC4J instance that you created.

➤ To deploy interop.war:

1 **In Oracle Enterprise Manager 10g Web console, select the** SharedServices **OC4J instance.**

2 **Click Application and then Deploy WAR file.**

3 **On the Deploy Web Application page, enter the following values to deploy the Web application:**

- **Web Application:** Click **Browse**, and select <*HYPERION_HOME*>/deployments/generic/SharedServices9/config/interop.war
- In **Application Name**, provide a name, for example, SharedServices.
- In **Map to URL**, type /interop.

4 **Click Finish.**

5 **On the Application: default page, click General to view the general properties.**

6 **Add the following libraries to the list of Library Paths:**

<*HYPERION_HOME*>/deployments/generic/SharedServices9/config;

<*HYPERION_HOME*>/deployments/generic/SharedServices9/config/lib/HubProductBean.jar;

<*HYPERION_HOME*>/common/JDBC/DataDirect/3.6/lib/hyjdbc.jar;

<*HYPERION_HOME*>/common/JakartaCommons/commons-collections-3.2.jar;

<*HYPERION_HOME*>/common/JakartaCommons/commons-dbcp-1.2.1.jar;

<*HYPERION_HOME*>/common/JakartaCommons/commons-logging-1.1.jar;

<*HYPERION_HOME*>/common/JakartaCommons/commons-pool-1.3.jar;

<*HYPERION_HOME*>/common/loggers/Log4j/1.2.14/lib/log4j-1.2.14.jar;

<*HYPERION_HOME*>/common/SAP/lib;

<*HYPERION_HOME*>/common/SAP/bin

<*HYPERION_HOME*>/common/XML/JAXP/1.2.2/xercesImpl.jar;

**Note:** Separate each value with a semicolon.

7 Click **Deploy**, and after the deployment process is complete, click **OK**.

8 Restart the `SharedServices` OC4J instance.

## Post-Deployment Tasks

➤ Complete these tasks after deploying:

1 Edit `<HYPERION_HOME>/deployments/generic/SharedServices9/config/CSS.xml` by modifying `<hub location>` with the correct hostname and port for your Oracle HTTP Server; for example:

```
<hub location="http://hostname.example.com:8888">
```

2 Restart the `SharedServices` OC4J instance.

# D

# Shared Services Backup and Recovery

**In This Appendix**



## Backing Up Shared Services

Backing up Shared Services includes backing up Shared Services application server configuration files, the OpenLDAP database configuration file, and all OpenLDAP data and log files. The Shared Services installation includes scripts that perform the backup process. Hyperion recommends backing up Shared Services data on a daily basis.

**Note:** Some Hyperion products require backup of additional data files. For example, for Oracle's Hyperion® Essbase® – System 9, you must back up an `essbase.sec` file. See the Hyperion product documentation.

Shared Services stores data in two databases:

- Relational database

  Shared Services supports the relational databases listed in the *Hyperion Installation Start Here*. The databases store event and administrator-related data. Procedures for backing up relational databases are specific to the type of database against whichShared Services is configured. See the database-vendor documentation for instructions.

- OpenLDAP database

  The OpenLDAP database is installed with and automatically configured by Shared Services. OpenLDAP stores security-services-related data.

**Note:** Shared Services can run during OpenLDAP backups.

To ensure that Shared Services can recover from catastrophic failure, back up data sources simultaneously (to ensure that some data is synchronized).

**Note:** If you performed a manual application server deployment (bypassing Oracle's Hyperion® Configuration Utility™), the backup script (<*HSS_HOME*>/server/scripts/hss_backup.sh) does not back up the application server configuration. Therefore you must manually edit the backup script with the correct path location of the Shared Services installation.

➤ To create a hot backup of OpenLDAP:

1 **Ensure that the Shared Services database is in online backup mode.**

2 **Run the** <*HSS_HOME*>/server/scripts/backup.sh *backup_folder_name* **command, where** <*HSS_HOME*> **is the location where Shared Services is installed and** *backup_folder_name* **is the path to the backup folder.**

Example:

/home/*username*/Hyperion/SharedServices/9.3.1/server/scripts/backup.sh /home/*username*/HSS_backup

3 **Optional: Copy the backup folder to a backup device, such as a CD-ROM, alternate disk, or tape.**

These files are backed up:

| Directory | Files |
|---|---|
| <*HYPERION_HOME*>/deployments/<*appServerAndVersion*><br>**Note:** For WebLogic 8.1.*x*, the location is <*HSS_HOME*>/AppServer/InstalledApps/WebLogic/8.1/. | Domain.xml<br>slide.properties<br>CSS.xml<br>WorkflowEngine.properties<br>Scheduler.properties<br>manage_data.properties |
| <*HSS_HOME*>/OpenLDAP | slapd.conf |
| <*HSS_HOME*>/OpenLDAP/var/openldap-data | *.bdb files<br>log.* files |

# Recovering Shared Services

To recover Shared Services from a catastrophic failure, you restore configuration and data files, and run the Sync OpenLDAP utility. The Shared Services installation includes scripts that perform the recovery process.

To recover Shared Services data, you recover the relational and OpenLDAP databases. When recovering from backups, ensure that the time stamps of the OpenLDAP database backup and the relational database backup match (or are close). The procedures for recovering relational databases are specific to the type of database against whichShared Services is configured. See the database-vendor documentation for instructions.

➤ To recover Shared Services:

1 **Perform an action:**

- For a normal (non-catastrophic) recovery, run the `<HSS_HOME>/server/scripts/recover.sh backup_folder_name` command, where `<HSS_HOME>` is the location where Shared Services is installed and *backup_folder_name* is the path to the backup folder.

  Example:

  `/home/username/Hyperion/SharedServices/9.3.1/server/scripts/recover.sh/home/username/HSS_backup`

  The recover script locates the backed up configuration and data files and places them in the appropriate directory under `<HSS_HOME>`. For the list of restored files, see "Backing Up Shared Services" on page 67.

- For a catastrophic recovery, run the `<HSS_HOME>/server/scripts/recover.sh backup_folder_name catRecovery` command, where `<HSS_HOME>` is the location where Shared Services is installed and *backup_folder_name* is the path to the backup folder.

  Example:

  `/home/username/Hyperion/SharedServices/9.3.1/server/scripts/recover.sh /home/username/HSS_backup catRecovery`

# Running the Sync OpenLDAP Utility

To ensure that Oracle's Hyperion® Shared Services relational and OpenLDAP databases are synchronized, run the Sync OpenLDAP utility. See *Hyperion Security Administration Guide*.

# Index

## S



## T



## U



## V



## W