Server Administration Guide

Oracle Utilities Smart Grid Gateway
Version 2.1.0.2 (OUAF 4.2.0.2.0)

E38619-02

April 2014

ORACLE

Server Administration Guide, Oracle Utilities Smart Grid Gateway, Version 2.1.0.2 (OUAF 4.2.0.2.0)

E38619-02

Primary Author: Oracle Tax And Utilities Global Business Unit

Table of Contents

Preface

Introduction

Welcome to the Oracle Utilities Smart Grid Gateway Server Administration Guide for Version 2.1.0.2. This guide outlines the technical concepts for operating and configuring the product on its platforms as outlined in the product installation documentation.

Note: All examples and screen captures are used for publishing purposes only and may vary from the actual values seen at your site.

Note: This document covers Oracle Utilities Application Framework V4.2.0.2.0.

Note: For publishing purposes, the Oracle Utilities Smart Grid Gateway product will be referred to as "product" in this document.

Note: All utilities in this guide are multi-platform (unless otherwise indicated). For publishing purposes the commands will be in the format ***`command[.sh]`*** *which indicates that the command can be used as* ***`command`*** *on the Windows platform or* ***`command.sh`*** *on the Linux/UNIX platforms.*

Note: Sections of this manual cover the batch aspects of the Oracle Utilities Application Framework for completeness only. Products that use the batch component of the Oracle Utilities Application Framework should refer to the dedicated Batch Server Administration Guide for specific advice about that component. Sections covering the Batch component are marked with a **BATCH** *graphic.*

Note: This document now covers aspects of the mobile framework used for mobile based products (Oracle Utilities Mobile Workforce Management and Oracle Realtime Scheduler). The relevant settings for this component are marked with a **MOB** *graphic. Not all products use the mobile framework; please refer to the installation guide for clarification.*

Note: This document now covers aspects of the bundling of Oracle Service Bus and SOA infrastructure. The relevant settings for this component are marked with **OSB** *and* **SOA** *graphics. Not all products use the Oracle Service Bus and SOA infrastructure natively; please refer to the installation guide for clarification. Refer to the Oracle SOA Suite documentation for more information about these products.*

Updates to This Documentation

This documentation is provided with the version of the product indicated. Additional and updated information about the operations and configuration of the product is available from the Knowledge Base section of My Oracle Support (http://support.oracle.com). Please refer to My Oracle Support for more information.

This document is regularly updated and should be re-downloaded on a regular basis. The Service Pack that applies to this document is indicated on the initial page of this document after the product version number.

Other Documentation

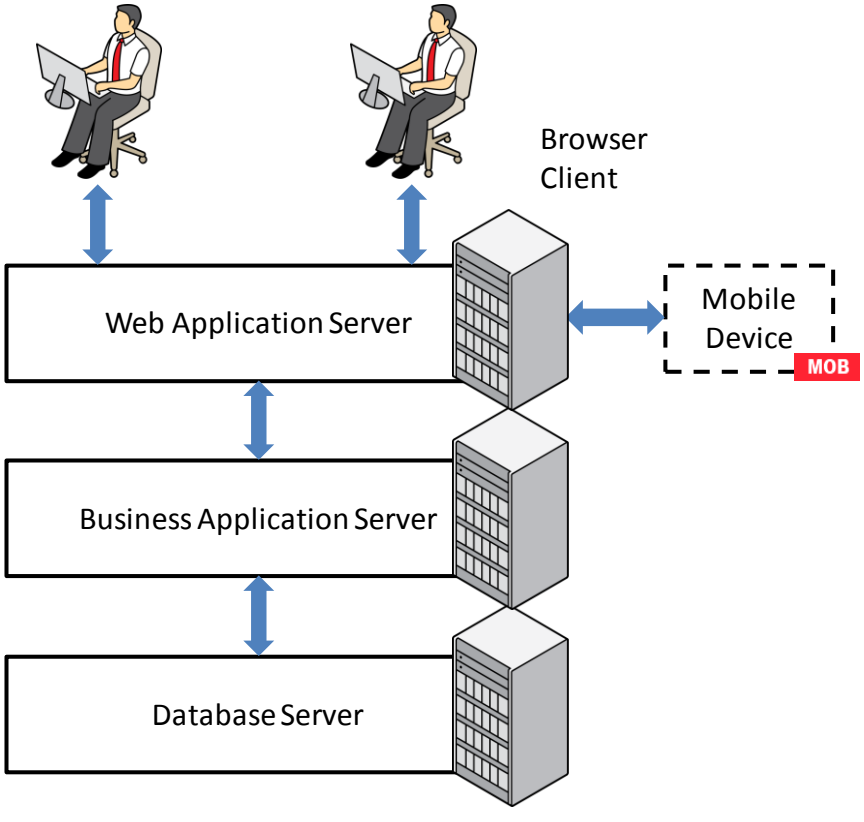
This document is part of the product technical documentation. There are groups of manuals that should also be read for additional specific advice and information:

- *Oracle Utilities Smart Grid Gateway Installation Guide*
- *Oracle Utilities Smart Grid Gateway Quick Installation Guide*
- *Oracle Utilities Smart Grid Gateway DBA Guide*
- *Oracle Utilities Smart Grid Gateway Security Guide*

These documents are available from http://edelivery.oracle.com

Architecture

The product is a multi-layered product with distinct tiers. The diagram below illustrates the architecture of the product:



The components of the architecture are as follows:

- **Browser Client** – The client component is a browser based interface which is *light* and only requires the Internet Explorer browser to operate.
- Communication between the client and server uses the HTTP protocol across a TCP/IP network. Secure Sockets (HTTPS) is also supported. The user simply uses a URL containing the product hostname and allocated port number in the address bar of Internet Explorer to access the application.

  Note: It is possible to use proxies to hide or translate the hostname and port numbers. Refer to the documentation provided with your J2EE Web application server documentation for proxy support instructions.
- **Mobile Device Terminal** – In some products the Mobile framework is deployed to allow mobile devices to interact with server processes. Refer to the product guides for applicability of the mobile framework to your product. MOB

  ***Note**: This manual has minimal information about the operation of the Mobile component of the Oracle Utilities Application Framework.*
- **Web Application Server (WAS)** – The product web application is housed in a J2EE compliant Web application server (Refer to the Supported Platforms section of the

installation guide for J2EE Web application servers and versions supported) This server can be run on a variety of supported Windows, Linux and Unix platforms (Refer to the Supported Platforms section of the guide for operating systems and versions supported). Within the Web application server the pages for the product are rendered using a combination of metadata and formatting rules to ensure a consistent look and feel. These pages are written using a combination of J2EE Java script and Java. These pages are cached on the Web Server and served to the client upon request. If the page requires business rules to be invoked then business objects are called from this server.

- **Business Application Server (BAS)** – The business component of the architecture can be installed as part of the Web application server (default) or as a separate component. This means the Business Application Server is also housed in a J2EE compliant Web application server (Refer to the Supported Platforms section of the installation guide for J2EE Web application servers and versions supported). This server can be run on a variety of supported Windows, Linux and Unix platforms (Refer to the Supported Platforms section of the installation guide for operating systems and versions supported). Within the Business Application Server the following components are implemented:
    - **Business Objects** – The business logic for each object in the system is expressed as a Java object. It contains all the SQL, programmatic rules and structures to manage the data for the transactions. In some products.
    - **DB Connection Pool** – If any database access is required, we use an industry component called Universal Connection Pool to manage and pool the connections to the database for the batch component and use the Web Server's own native JDBC connection pooling for the online and Web Services component. This will reserve connections and ensure efficient use of connections to the database. To access the database product uses the networking client provided by the DBMS vendors to ensure correct connection. For example, Oracle provides SQL*NET, DB2 provides UDB Connect and SQL Server uses .NET drivers. These clients are multi-protocol for maximum flexibility.
- **Database Server** – The RDBMS used for the implementation is implemented in the database server. The product supports a number of databases (Refer to the Supported Platforms section of the installation guide for databases and versions supported). The database server only stores and retrieves the data for the product as all the business logic is in the business objects.

Roles and Features

Each tier in the architecture has a specific role in the operation of the product. The sections below outline the roles and features of each tier.

Client

The Browser User interface (BUI) is a combination of HTML and Java-script. AJAX, shorthand for Asynchronous JavaScript and XML, is a Web development technique for creating interactive Web applications. This makes web pages more responsive by exchanging small amounts of data with the server, so that the entire page does not have to be reloaded each time the user makes a change. This increases the Web page's interactivity, speed, and usability.

Note: Refer to the installation guide for the supported browsers and the supported versions of those browsers.

There are no ActiveX or Java components in the base product installation. This means that the deployment of the browser client is relatively simple as the only required component to use the product is a supported version of Internet Explorer on the client machine. If the implementation requires ActiveX controls for extensions then they can be added and used for the implementation.

Note: If your implementation chooses to use the graphing component zones, then the latest version of the Macromedia Flash browser component must be installed. Refer to http://www.adobe.com/products/flashplayer/

The Browser tier of the product is provided for the end users to access the product on a desktop. The client provides the following roles in the architecture:

- **Screen Rendering and Caching** – All the screens are rendered using standard HTML and JavaScript (not Java). The rendering is performed as the screen is served from the Web Application server and stored in the local browser cache.
- **User Interaction** – The client provides the user with the screen interaction. After page is rendered the user can interact (manipulate data and screen elements) as per their business transaction. The browser client is responsible for ensuring that users can navigate and interact with the screen elements (e.g. resizing, display correctly).
- **User Context** – The product is stateless and therefore the client stores the transactional context locally and passes this to the transaction as required. The client records the context of the transaction in the browser memory.

No business logic is stored on the client component.

Web Application Server

The product is a J2EE set of Web applications that are housed in a J2EE compliant Web application server. The product and the Web application server provide the following roles in the architecture:

- **Authentication** – The Web application server software that houses the product provides adapters to common security repositories. This means that security products interfaced to the Web application server software can be used in conjunction (with configuration) with the product.
- **Managing Client connections** – The Web application server software manages any

client connections (during and after they are authenticated) for processing and availability.

- **Page Server** – The major responsibility of the Web application server is to *serve* pages to the client on demand. At start-up time (or at the first request for a particular page) the product generates the screens dynamically using metadata and rendering style sheets. These are cached for reuse locally.
- **Cache Management** – For performance reasons, the static data (usually metadata and configuration data) is cached in memory on the Web application server.

No business logic is stored on the Web application server component. The Web application server Component of the product is written in Java and JavaScript.

Business Application Server

The product is a J2EE set of business applications that are housed in a J2EE compliant Web application server (this can be the same instance of the Web application server or a separate one). The product and the Business Application Server provide the following roles in the architecture:

- **Authorization** – After authentication has been performed by the Web application server, the Business Application server is responsible for determining which functions and which data can be accessed.
- **Data Integrity** – The Business Application Server contains the business logic to maintain referential integrity for the product data.
- **Validation** – The Business Application Server contains the business logic that contains all the validation rules for the product data.
- **Business Rules** – The Business Application Server contains the business logic that implements business rules and performs calculations.
- **SQL** – The Business Application Server contains all the SQL statements and formats and processes results from those SQL statements.

The Business Application Server Component of the product is written in Java.

Database Server

The product contains a database schema within a database management system. The database server has the following roles in the architecture:

- **Data Storage** – The database is responsible for efficiently storing all data.
- **Data Retrieval** – The database is responsible for efficiently retrieving data using SQL provided by the Business Application Server.
- **Data Management** – The database is responsible for efficiently managing all data.

No business logic is stored on the Database Server.

Concepts

Before you attempt to configure or operate the product, there are important concepts that you should understand. These concepts are addressed in this document as a basis for the other documents in the technical documentation.

Environment

In a product implementation and post-implementation there will be a number of copies of the product installed. Each copy of the product is known as an environment. Each environment will be created for a specific purpose, according to your site plans, and accessible to a group of users deemed necessary for that purpose. For example, there will be at least one testing environment where designated personnel will perform their testing tasks.

For planning purposes an environment is an instance of:

- The Web applications deployed in a J2EE Web application server. This may be in a single server or across a cluster of servers.
- The business applications deployed in a J2EE Web application server. This can be the same physical J2EE Web application server or another instance (such as a separate server). This may be in a single server or across a cluster of servers.
- A database containing the product schema. Physically, a schema can exist in an individual database instance or shared within a database instance (i.e. you can install multiple schemas of the product in the same database). This may be in a single server or across a cluster of servers.

While there is no restriction on the number of environments it is recommended that the minimal number of copies of the product is installed using the guidelines outlined in the Environment Management document in the Software Configuration Management series KB Id: **560401.1** on My Oracle Support.

Administration User Id and Group

Prior to installing the product, you create a UNIX administration user ID and administration group. This account is used to install and operate the product. The product administration user ID and product group is provided as a parameter during the installation process. By default, the product administration user ID is **splsys** (**SPLADMIN** parameter and environment variable) and the group is **splusr** (**SPLADMINGRP** parameter and environment variable). However, alternative values can be used according to your site standards.

The administration userid is responsible for the following:

- It is the owner of the majority of the files installed for the product.
- It is the only userid that should be used to run any of the administration tools provided with the product.
- It is the userid that owns the UNIX resources used by the product. When the product

is running, this userid owns the processes associated with running the base software.

The administration userid should be protected from unauthorized use. If components of the responsibility of administration need to be delegated to other users on the machine, we recommend not giving out the administration userid. Instead, an alternative solution may be sought (such as using *sudo* or similar security tools).

The administration userid should not be used for any of the following:

- As a product end user. By default, the administration userid does not have access to the functionality of the product.
- To run product background processes.
- To manipulate data files exported from or imported into the product from any interfaces.

This technical document will refer to the administration userid as **splsys**. If your site uses an alternative userid as the administration userid, substitute that userid value for **splsys**.

Implementation Tip: It is possible to implement a different owner per environment in the product. Why would you want to do this? If you want to allow developers or testers to restart environments themselves, you can give access only to appropriate environments to distribute the administration. This can be achieved by installing the product with different userids. You must log in and administrate each environment with its account only.

Native Support vs Embedded Support

Note: This facility applies to Oracle WebLogic customers only. IBM WebSphere uses Native Support only.

By default, the utilities and configuration files use Oracle WebLogic in *embedded* mode. In this mode the Oracle WebLogic installation does not house the deployment of the product within the Oracle WebLogic installation structure. The utilities and configuration files allow the Oracle WebLogic installation to logically reference the structures and deployment files from the `splapp` directory within the product installation. The process uses templates to create product configuration files as well as Oracle WebLogic configuration files and utilities to logically reference the files in the product structure.

This has advantages where a single installation of Oracle WebLogic can be used for multiple product environments (including development environments and different products on the same framework) and is therefore ideal for non-production environments.

Whilst the embedded environment is recommended for non-production environment it has limitations that are not practical for a production environment. These limitations are:

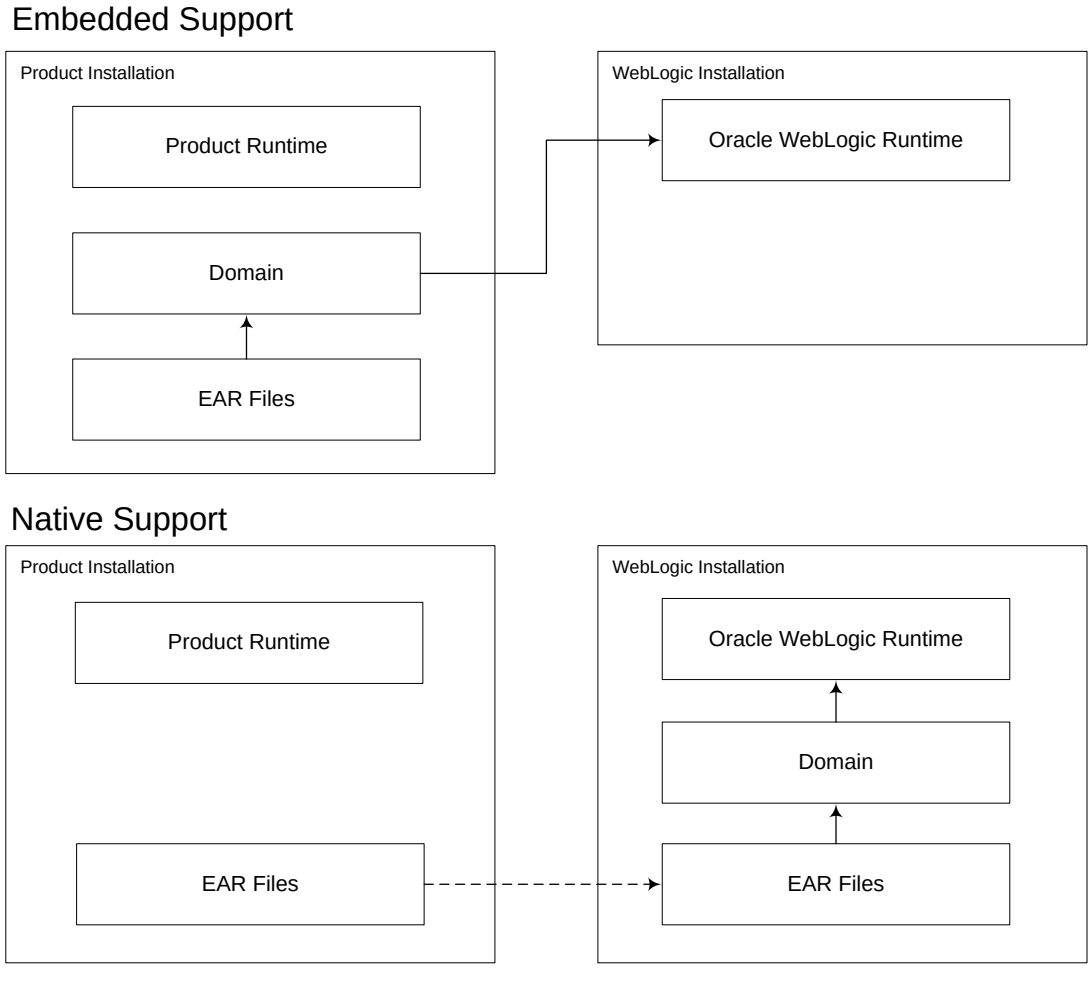
- **Clustering Support** – By default the installation and creation of the configuration files predefines a simple installation with a single server. In production Oracle WebLogic clustering is typically used and this is difficult to configure and maintain when using *embedded* mode, without manual manipulation of configuration files.
- **Administration Server installation** – In each installation of the product the administration console is deployed in each server which is not recommended for

production environments. Typically, a single installation of the administration server will exist, or Oracle Enterprise Manager will be used, for production management of the environment.

- **Set configuration** – The product installation contains a set configuration which is a common installation for non-production environments and is not optimized for production.

It is possible to use the *native* mode of Oracle WebLogic to house the product. In this case, the product is installed as outlined in the Installation Guide but deployed to the Oracle WebLogic domain location using the Oracle WebLogic deployment tools and managed from the console. This allows the native tools to be used instead of the provided utilities and allows for the console to be used to manage the product. In this mode the product runtime for the product is either embedded in the EAR files or referenced indirectly by the EAR file.

The figures below illustrate the architecture differences in the two approaches:



In the *embedded* approach the domain files exist under the product installation and are built and maintained using the `configureEnv` and `initialSetup` utilities. No product installation files exist in the Oracle WebLogic installation but the Oracle WebLogic runtime is used by the generated utilities. Hence the term, *embedded* mode. This is recommended for non-production environments as it minimizes the number of Oracle WebLogic installations.

In the *native* approach, the domain files exist under the Oracle WebLogic installation location, as other Oracle applications use. The product files are deployed to the Oracle

WebLogic location using the deployment utilities provided by Oracle WebLogic (console or WLST can be used). This means that all the administration for any Oracle WebLogic configuration can be performed from the console rather than from command lines and configuration files. This approach is recommended for customers using Oracle ExaLogic and/or Oracle WebLogic clustering.

In both modes all the product specific configuration files are maintained using the facilities outlined in this manual. Refer to the Installation Guide for steps to setup *embedded* or *native* mode.

Note: For native installation instructions refer to Native Installation Oracle Utilities Application Framework (Doc Id: 1544969.1) available from My Oracle Support.

This guide will outline the operations and configuration for both approaches.

Directory Structure

In an effort to facilitate upgrades and ease maintenance, the product installation process creates a very specific directory hierarchy under the administration user ID of **splsys** (by default). The structure holds all the code, system products, scripts and temporary files that are created by the product during installation and operation.

Note. Every part of the product relies on the fact that this directory structure and the files within remain intact as delivered.

Note. At no time should you modify any of the supplied programs or scripts without the express direction of Oracle

There are two different directory structures that the product application uses:

- Base code directory structure (denoted in this documentation as ***<SPLDIR>***)
- Application output directory structure / log directory (denoted in this documentation as ***<SPLDIROUT>***)

Within each of the structures, there is a mount point and a subdirectory for each environment <environment> installed on the machine. The base mount point ***<SPLDIR>*** contains the environment directories that hold all of the application software for each particular environment. The application output mount point ***<SPLDIROUT>*** contains the environment directories that hold temporary files (such as the output batch) as well as batch log files. The default ***<SPLDIR>*** directory is **`/spl`** and the default ***<SPLDIROUT>*** directory is **`/spl/sploutput`**.

When a user logs on to a particular environment of the product either using the browser-based interface or directly on UNIX/Windows, the environment is set up (i.e. environment variables, etc.) to point to the appropriate directory structure under the mount point. The environment variable that points to an environment directory under ***<SPLDIR>*** is **`$SPLEBASE`** (or **`%SPLEBASE%`** in Windows). The environment variable that points to an environment directory under ***<SPLDIROUT>*** is **`$SPLOUTPUT`** (or **`%SPLOUTPUT%`** on Windows). The **`SPLEBASE`** and **`SPLOUTPUT`** environment variables are two of the standard environment variables used by the utilities provided with the product and runtime.

Implementation Tip. The actual location of the application directory ***<SPLDIR>*** *and application output directory* ***<SPLDIROUT>*** *is up to site standards. The product does not care where it is installed as it internally uses the environment variables to access the correct locations.*

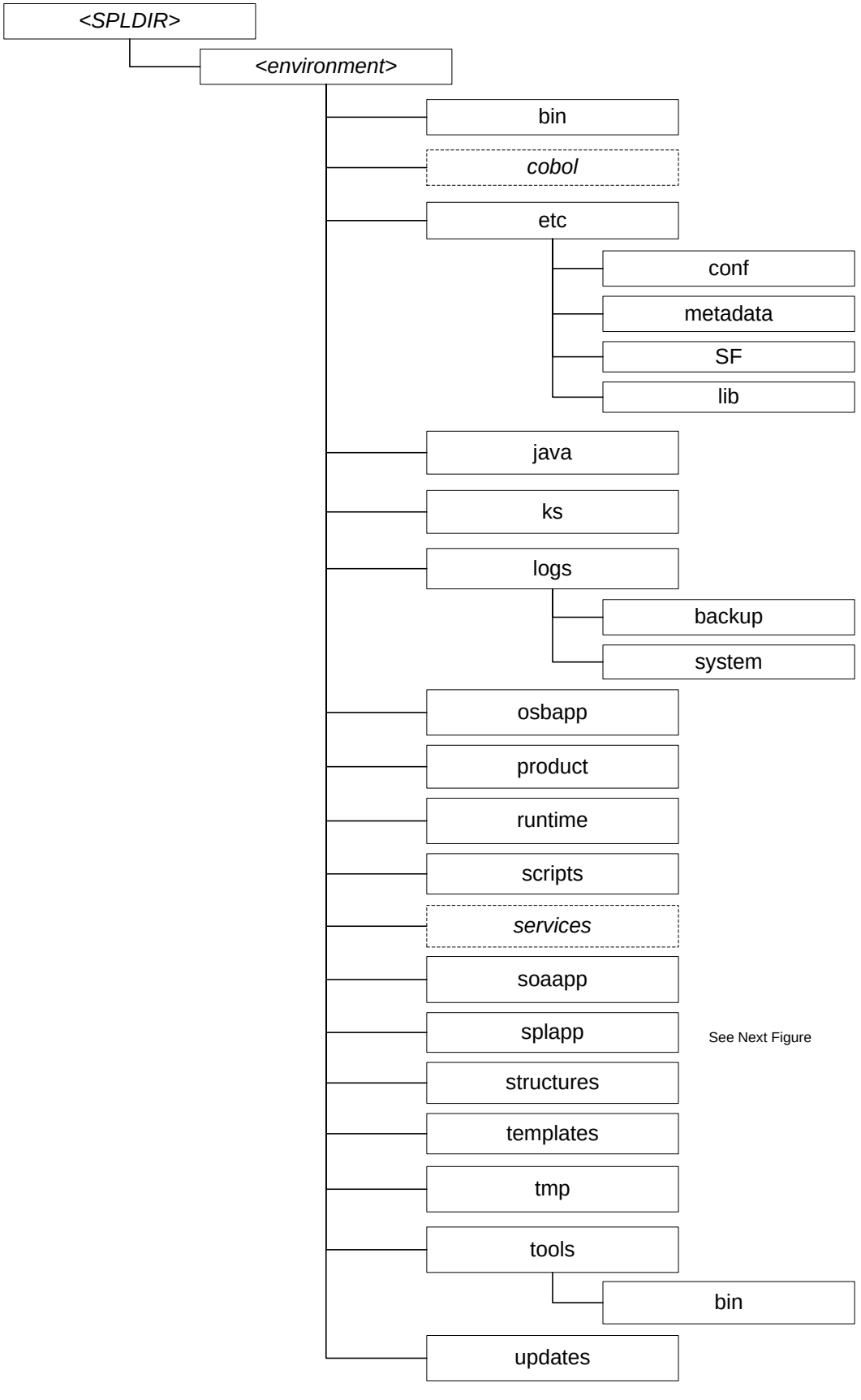
The actual location for the mount points can differ per environment if you want. This is handy if you need to vary the location because you do not have enough space for all your non-production environments. Typically the number of environments during an implementation varies according to the level of access and desired amount of testing and training. The only restriction is that there can only be one location for **SPLEBASE** and **SPLOUTPUT** per environment.

Software Directory Structure

The following components are stored in the base code directory structure:

- **Runtimes for Components** – All the runtime executables for the base software.
- **Business Object Binaries** – All the binaries that contain the business logic.
- **Configuration Files** – All the configuration files for the business objects and runtimes
- **Scripts** – Any administration or runtime scripts that are supplied to the customer.
- **Supported Plug-ins** – Source and executable for supplied plug-ins.

The following figure depicts the layout of where the product code is placed upon installation into the file system (where ***<environment>*** is the environment name chosen during the installation process):

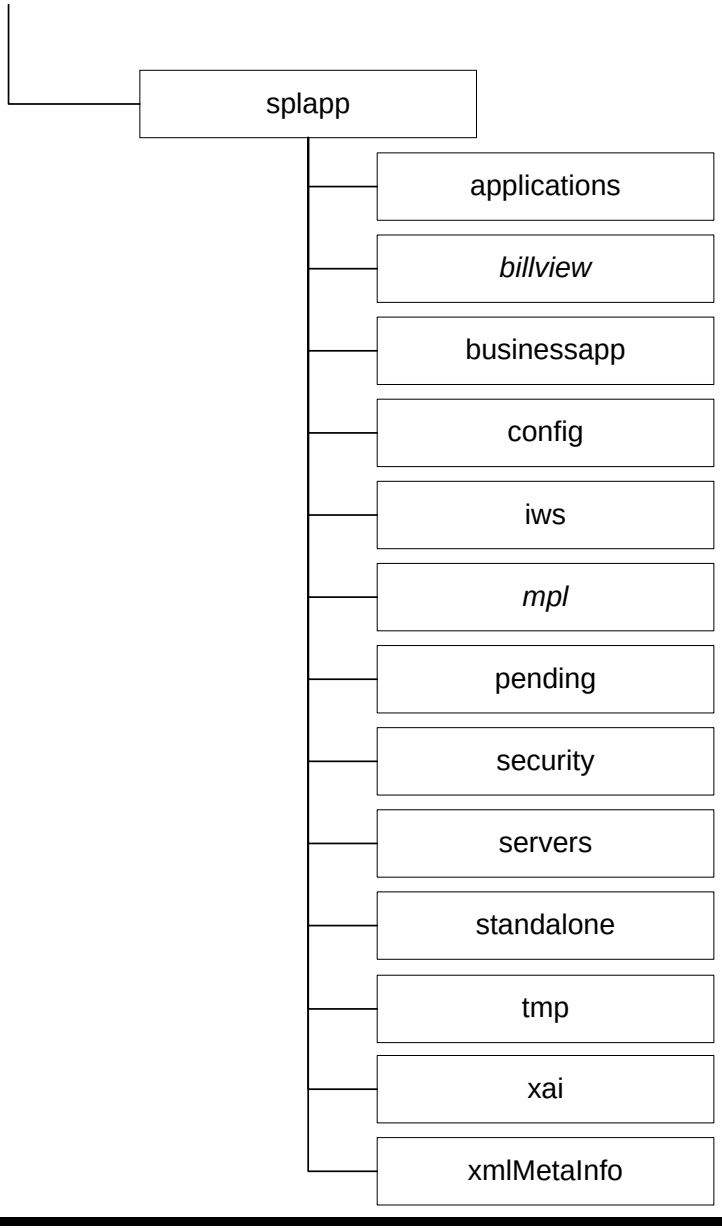


The following table outlines the typical contents of these directories:

Directory	Contents

Directory	Contents
bin	Utilities and commands for operations and configuration.
cobol	For products that support COBOL, a set of subdirectories that contain the source and object code for any supplied COBOL based plug-ins. Any compile output is also held in this structure. The source directory can be referenced by the environment variable **SPLSOURCE**. The build directory can be referenced by the environment variable **SPLBUILD**.
etc	A set of directories holding configuration files used in the product as well as template files and base libraries used to generate the configuration files.
java	Location of temporary files for java execution
ks	Location of Key Stores
osbapp	Oracle Service Bus integration (*optional*)
product	Directories containing any bundled software with the product.
runtime	Directory containing any compiled objects for the product.
scripts	Directory containing any implementation specific scripts.
services	For products that support COBOL, directory containing COBOL source service definitions for the development kit and compilation
soaapp	Oracle SOA integration (*optional*)
splapp	Directories containing the J2EE Web Applications (see below)
structures	Internal structures used for configuration utilities
templates	Base templates used to build configuration files
tmp	Directory used to hold intermediary files used for the deployment process
tools	Location of service pack, single fix and group fix utilities
updates	Location of installed_fixes.txt

Under the **splapp** subdirectory for each environment there are a number of subdirectories:



Directory	Contents
applications	Location of the Web application product files
billview	Location of the online bill viewing files (Products supporting bill view only)
businessapp	Location of the business application product files
config	Location of temporary configuration files.
iws	Location of Native Web Services
mpl	*Location of Multi-Purpose Listener (selected products)*
pending	Location of build temporary files
security	Default location of domain security initialization files (Oracle WebLogic only)
servers	Default location of copies of configuration and associated files (Oracle

Directory	Contents
	WebLogic only)
standalone	Location of common Java libraries and the batch component of the product. Used for batch component.
tmp	Temporary directory used in build process
xai	Location of the Web services adapter configuration and Incoming service schemas
xmlMetaInfo	Location of the service definitions for the product.

Warning: Under no circumstances should files be manually altered in these directories unless instructed by Oracle Support. The Oracle Utilities SDK will deposit files in the relevant locations in this structure using the Packaging component of the SDK or using the Development tools directly

Directory Permissions

Note: This facility on applies to Linux and Unix platforms only.

The directories within the product are controlled by the operating system security relating to the administration user assigned to the product. The table below outlines the permissions under the **$SPLEBASE** location:

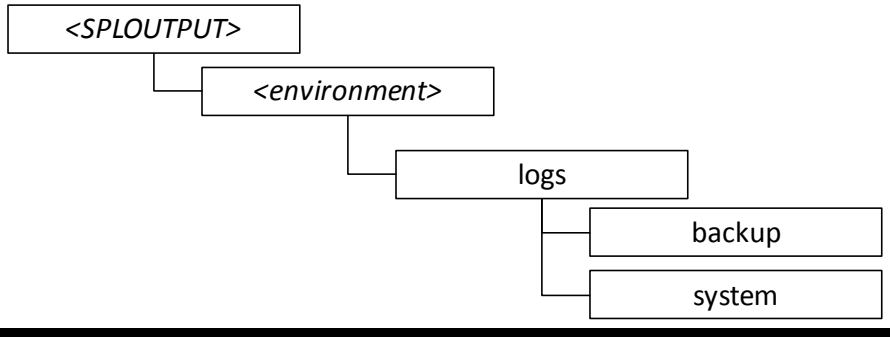
Directory/files	Owner	Group	Other
cobol	rw	R	
etc	rw	r	
Individual files	rw	r	
logs	rw	r	
logs/system	rw	rw	
Oracle WebLogic Configuration Files	rwx	rw	
Oracle WebLogic shell scripts (.sh)	rwx	rx	
product	rwx	rx	
product/apache-ant/bin/ant	rwx	rx	rx
runtime	rwx	r	
scripts, tools, updates	rwx	rx	
Shell scripts (.sh) in bin	rwx	rx	
osbapp, soaapp, splapp	rwx	rx	
structures	rwx	rx	
templates	rwx	rx	
tmp	rwx	rx	

Legend: r = Read Only, rw = Read Write, rwx = Read Write Execute, rx – Read/Execute, blank = no permissions.

These permissions are set by the **setpermissions.sh** utility which is executed as outlined in the Installation Guide.

Output Structure

The product processes (batch and online) that produce output and logs place information in this directory structure. The environment directories are referenced by the environment variable **SPLOUTPUT**. By default, this directory is created as **/spl/splapp**, though this can be substituted for an alternative during the installation process. The figure below illustrates the typical directory structure for this location (where ***<environment>*** is the environment name chosen during the installation process):



Directory	Contents
logs/system	Directory containing application logs files. This is independent of Web application server, Business Application Server and Database Server log files.
logs/backup	Default location of backup of logs if **SPLBCKLOGDIR** is set.

The implementation may add subdirectories as their site standards and implementation dictates.

Environment Variables

The product uses a number of environment variables to determine where information is stored and to be placed for its internal operations. Becoming familiar with these variables will assist you in finding information quickly and efficiently.

Note. If a custom script is written to access or write information to the product functionality, it is highly recommended that the following variables be referenced in your scripts. This is to maximize the chance that your script will remain functional across upgrades.

The following table outlines some of the key environment variables:

Variable	Usage
ADF_HOME	Location of the Oracle ADF files
ANT_ADDITIONAL_OPT	Options for *ant* as per the configuration settings at

Variable	Usage
	installation time
ANT_HOME	Location of *ant* build utilities
ANT_OPTS	Options for *ant* for Oracle SDK
ANT_OPT_MAX	Maximum memory settings for *ant* as per the configuration settings at installation time
ANT_OPT_MIN	Minimum memory settings for *ant* as per the configuration settings at installation time
CMPDB	Database Type (**ORACLE**)
CHILD_JVM_JAVA_HOME	Location of JVM used for COBOL integration *(COBOL based products only)*
COBDIR	Location of COBOL runtime *(COBOL based products only)*
COBJVM	Name of JVM for COBOL integration *(COBOL based products only).*
COBMODE	Execution mode of COBOL runtime (32 or 64 bit) *(COBOL based products only)*
CUSTCOBDIR	Location of custom COBOL installation (if used) *(COBOL based products only)*
DBNAME	Database Name
ENVFILE	Location and name of environment configuration file
HIBERNATE_JAR_DIR	Location of Hibernate Java library
HOSTNAME	Name of host
JAVA_HOME	Location of JDK
LANG	Language for COBOL *(COBOL based products only)*
LC_MESSAGES	Messages for COBOL *(COBOL based products only)*
NLS_DATE_FORMAT	Oracle NLS Date Format
NLS_LANG	Oracle NLS Language string
NLS_SORT	Oracle NLS Sorting
ONLINEBILLINI	Location of document rendering software template building configuration files
ONLINEDOCINI	Location of document rendering software configuration files
OPSYS	Operating System Name
OPSYSINFO/OPSYSVER	Operating System Version
ORACLE_CLIENT_HOME	Location of the Oracle Client software (used for location of perl). If full Oracle Database software is installed on the host

Variable	Usage
	this value will match **ORACLE_HOME**.
ORACLE_HOME	Location of the DBMS software
PERL5LIB/PERLIB	Location of Perl Libraries
PERL_HOME	Location of Perl
SPLADMIN	Administration user ID
SPLADMINGROUP	Administration group
SPLApp	Name of root Web application WAR file.
SPLAPP	See **SPLOUTPUT**
SPLBCKLOGDIR	Location of backups of online log files (must be set manually).
SPLBUILD	Location of COBOL build directory *(COBOL based products only)*
SPLCOBCPY	Location of COBOL copy code libraries *(COBOL based products only)*
SPLCOMP	Name of COBOL compiler vendor *(COBOL based products only)*
SPLEBASE	Location of software for environment
SPLENVIRON	Name of environment
SPLOUTPUT	Location of output for environment
SPLRUN	Location of runtime for environment
SPLSDKROOT	Location of SDK (Development environment only)
SPLSOURCE	Location of COBOL source *(COBOL based products only)*
SPLSYSTEMLOGS	Location of product specific logs
SPLVERSION	Version identifier of product (prefixed with *V*)
SPLVERSION_NUM	Version number of product
SPLWAS	Web application Server type
WEB_ISEXPANDED	Whether Web application is expanded or not (not = WAR/EAR files)
WEB_SERVER_HOME	Location of Web Application Server software
WL_HOME	Location of Oracle WebLogic installation (WebLogic supported platforms only)
XAIApp	Name of the XAI Application WAR file

Note: If a custom script is written to access or write information to the product functionality, it is highly recommended that the following variables be referenced in your scripts. This is to maximize the

chance that your script will remain functional across upgrades.

*Note: **HIBERNATE_JAR_DIR** is used for the installation process only. After installation is complete the jar files located at the locations specified by these environment variables are copied to the correct locations for execution.*

Common Application Logs

When the product is operating the infrastructure logs messages within its own logs. For example, the database will log database errors or messages to the database logs, the J2EE Web application server will log Web Application errors or messages to the J2EE Web application server logs and so on. The name and location of these logs is set by relevant vendors of those logs. Refer to the documentation provided with that software on where logs are stored and their logging conventions.

The product additionally writes a number of application specific logs to **$SPLSYSTEMLOGS** (or **%SPLSYSTEMLOGS%** on Windows):

- spl_web.log - Web application server application messages .
- spl_service.log – Business Application Server messages. If the Business Application Server exists on the same J2EE Web Application Server instance (i.e. as per a *local install*) as the Web application server for an environment then this log does not exist and all messages are written to the **spl_web.log**.
- spl_xai.log – Web Services Adapter messages.

The format of all logs is as follows:

Field	Comments
<userid>	User ID of transaction (blank or "-" for system generated messages)
<pid>	Process identifier (optional)
<time>	Time of transaction in format HH:MM:SS,SSS
[<transaction>]	Transaction/Class identifier
<type>	Type of message
(<class>)	Java class generating message (see Javadocs in appViewer)
<message>	<message contents>

Sample log entries:

```
19:03:16,390 [main] INFO  (support.context.CacheManager) Registering cache
'MenuRepository'

-  19:02:37,812 [main] INFO  (support.context.ContextFactory) 461 services
registered, time 11.742 ms

- 19:03:29,140 [Remote JVM:2 Thread 1] WARN  (cobol.mem.CobolModeHelper)
Unspecified or unrecognized COBMODE (null) - inspecting JVM properties to
determine bit mode ...

19:03:40,875 [Thread-24] ERROR (web.dynamicui.MetaDataHolder) Unable to
find UI xml file '/an/generated/toDoSummaryListGrid.xml' for program
'toDoSummaryListGrid'
```

```
DEMO - 259992-101-1 19:17:38,750 [http-6500-5] INFO
(support.context.CacheManager) Registering cache 'UiMapInfoCache'
```

Automated Backup of Log files

Note: This facility is only supported using the embedded mode of Oracle WebLogic. If native mode is used then Oracle WebLogic log management is used.

When the product is started with the spl command and if the **SPLBCKLOGDIR** environment is set then the logs are backed up to the location specified in the **SPLBCKLOGDIR** environment variable. If this variable is not set then the logs are removed prior to the start of the product.

The logs are backed up with the following pattern:

$SPLBCKLOGDIR/*<datetime>.<SPLENVIRON>.<logfilename>* (Linux/UNIX)

or

%SPLBCKLOGDIR%*<datetime>.<SPLENVIRON>.<logfilename>* (Windows)

where

<datetime>	The date and time of the backup in the format YYYYMMDD.HHMM
<SPLENVIRON>	The environment name
<logfilename>	The name of the original file that is backed up.

Attaching to an Environment

Note: This command is not necessary if using the Oracle WebLogic native support when managing the product from the Oracle WebLogic console.

Before performing any command against a product environment, you must attach to the environment. Attaching to an environment sets system and environment variables so that the correct runtime and code is used in the execution of subsequent commands.

To attach to an environment:

- Make sure that you are logged in using the administration account for the desired environment, for example **splsys**.
- Execute the following command:

```
<SPLDIR>/<environment>/bin/splenviron.sh –e <environment>
```

Or

```
<SPLDIR>\<environment>\bin\splenviron.cmd –e <environment>
```

Where ***<SPLDIR>*** is the mount point defined for the product and <environment> is the name of the environment to access.

Note: This command must be run before any UNIX-based command (including running the product background processes) to ensure that the correct environment is in place.

*Note: If you are running multiple versions of the product, ensure that you run the correct version of the **splenviron[.sh]** utility for the environment by manually changing to the directory where the **splenviron[.sh]** utility exists for the desired environment prior to running the command.*

The following is an example of splenviron.sh execution:

```
$ /spl/DEMO/bin/splenviron.sh -e DEMO

Version ................ (SPLVERSION) : Vx.x.x
Environment Name ....... (SPLENVIRON) : DEMO
Environment Code Directory (SPLEBASE) : /spl/DEMO
App Output Dir - Logs ... (SPLOUTPUT) : /spl/sploutput/DEMO
Build Directory ......... (SPLBUILD) : /spl/DEMO/cobol/build
Runtime Directory ......... (SPLRUN) : /spl/DEMO/runtime
Cobol Copy Path ......... (SPLCOBCPY) :
```

*Note: The value of **SPLCOBCPY** is only shown for products supporting COBOL.*

The above example summary of the command illustrates that important environment variables and their values are set. Use this information to confirm that you have successfully attached to the correct environment.

Utilities

The product includes several command scripts to aid with its configuration and operation. This section provides information about these utilities.

splenviron – Set Environment variables

Note: On the Linux/UNIX environment this utility creates a subshell upon completion.

The **splenviron[.sh]** utility initializes a defined set of environment variables and paths for an environment. This script must be run before any other script or utility is run within the environment.

Command Usage:

Linux/Unix:

```
splenviron.sh -e <environment> [-c <command>] [-q] [-h]
```

Windows:

```
splenviron.cmd -e <environment> [-c <command>] [-q] [-h]
```

Where:

-e *<environment>*	*<environment>* is the environment id as installed in the **cistab** file.
-c *<command>*	Execute *<command>* after running **splenviron[.sh]**. Command must be enclosed in double quotes (""). Default is shell (e.g. ksh).
-q	Quiet Mode. Do not show output from command. Any output from

	the –c command will be shown.
`-h`	Show usage.

Samples:

```
splenviron.sh –e DEMO
splenviron –e DEV
splenviron.sh –e DEMO –c "cat file.lst"
```

The **splenviron[.sh]** utility is executed whenever an environment needs to be initialized. One of the options to this script allows system administrators to optionally include the execution of an additional command as part of the environment initialization. This enables the system administrator to more finely tune the environment shell so they can change such settings as TimeZone, PATH or environment variables.

Extending the splenviron Command

If your implementation needs to add environment variables (or modify existing variables) for a third party product you may wish to integrate with that product. For example, you might want to add some custom Java classes from a component that you want to use with the product.

When you run the **splenviron[.sh]** utility it sets the environment variables for the environment. These are standard variables as well as any required for operation of the product. For example, there are variables that can be used in utilities so they can be used across environments.

These environment variables can be extended (or added to) using one of the following options:

- **Change to ALL environments on machine** - If your integration is common across all environments then you can set or alter environment variables using the following technique:
  - Create a script in a central location on the machine that sets or alters the appropriate environment variables. Ensure that the product administrator user ID has read/execute access to the location and the script.
  - Set the **CMENV** environment variable with the location and name of the script to execute prior to running the **splenviron[.sh]** utility (for example, in your logon profile).
  - When the **splenviron[.sh]** utility is run it will detect the script specified in the **CMENV** environment variable and execute the script to set or alter the environment variables.
- **Change to a specific environment on machine** - If your integration is specific to an environment (or different for each environment, for example if you have a development as well as a test copy of the third party product) then you can set or alter environment variables using the following technique:
  - Create a script called **cmenv.sh** (or **cmenv.cmd** on Windows) in scripts

subdirectory of the environment (usually **$SPLEBASE/scripts** or **%SPLEBASE%\scripts**). Ensure the permissions are set appropriately for the product administration account to execute the script.

- When the **splenviron[.sh]** utility is run it will detect the **cmenv.sh** script (or **cmenv.cmd** on Windows) and execute the script to set or alter the environment variables at the end of the **splenviron[.sh]** utility.

- Combination of both previously outlined options – It is possible to combine the techniques in a combination which can mean you can have maximum flexibility. If you follow the instruction of both techniques then the following will happen in the following order:
  - When the **splenviron[.sh]** utility is run it will detect the script specified in the **CMENV** environment variable and execute the script to set or alter the environment variables.
  - If there is a **cmenv.sh** script (or **cmenv.cmd** on Windows) in the scripts subdirectory of the environment, it will execute the script to set or alter the environment variables. This may override, add or alter environment variables already set.

In using this override technique, remember:

- If you alter any pre-existing environment variables then ensure your changes are not going to circumvent product requirements. For example, do not alter paths used by the product.
- If you add files or directories to library variables or **CLASSPATH** ensure your changes are suffixed at the end of the variable. This is especially important for java classes as classes you use may conflict with product supplied ones; adding them at the end of the **CLASSPATH** will minimize the effects of conflicts.
- Do not remove any environment variables used by the product.

configureEnv – Setup Environment settings

Note: This utility can be used by both embedded and native mode customers. In native mode, some settings need to be specific values to support the native mode. Refer to the Installation Guide for further instructions on the use for the different modes.

The **configureEnv[.sh]** utility is an interactive method for configuring an environment on the system stored in the **etc/ENVIRON.INI**. This configuration script sets up important parameters used by other scripts within the system. Normally this script is executed without parameters and the current environment (i.e., the environment that you are currently attached to) is configured.

Command Usage:

Linux/Unix:

```
configureEnv.sh ([-a]|[-g]) [-i] [-h]
```

Windows:

```
configureEnv.cmd ([-a]|[-g]) [-i] [-h]
```

Where:

blank	Configure basic configuration options
-a	Configure advanced configuration options
-g	Configure all configuration options (basic and advanced).
-h	Show usage.
-i	Configure Installation options (used for initial installation)

Refer to ENVIRON.INI for more information on the output of this command.

*Note: If an unauthorized user attempts to execute this command the following error message – "**Can't open …./configure.log for output**" is output.*

spl – Start/Stop Environment

*Note: The **splenviron[.sh]** utility must be executed before this utility can be used. See splenviron – Set Environment variables for details.*

Note: This utility should not be used for native mode customers. Use the console or scripts supplied with Oracle WebLogic to start or stop the product. Refer to the Oracle WebLogic documentation for information on this capability.

The **spl[.sh]** utility is used to start up and shut down an environment or individual components (web server or multi-purpose listener) of an environment. Usage of this utility is optional in sections of this document.

Use the command without a parameter to start up, reboot or shut down all components of an environment (note that the action must still be used). To start up or shut down an individual component, use the option that specifies that applies to that specific component.

Command Usage:

Linux/Unix:

```
spl.sh [-h] [-wsba] [-q] <action>
```

Windows:

```
spl.cmd [-h] [-wsba] [-q] <action>
```

Where:

-h	Show usage.
blank	Perform ***<action>*** on Web application server/Business Application only
-w	Perform ***<action>*** on Web application server only
-s	Perform ***<action>*** on Business application server only
-b	Perform ***<action>*** on batch component only. DEFAULT threadpool only. BATCH

`-m`	Perform *`<action>`* on MPL only (selected products only)
`-a`	Perform *`<action>`* on all components
`-q`	Quiet Mode – Non-critical output goes to log file only
`<action>`	**`start`** – start the component/environment
	`stop` – stop the component/environment
	`check` – Check the status of the environment

When executed the script returns the following return codes:

Return Code ($?)	Comments
0	Command executed successfully
1	Command executed unsuccessfully

Note: The command may issue other commands that need to be tracked separately depending on the platform. For Example

Action	Linux/Unix Command	Windows Command
Start Application Server	`spl.sh start`	`spl start`
Stop Application Server	`spl.sh stop`	`spl stop`
Start all components	`spl.sh –a start`	`spl –a start`
Stop DEFAULT threadpool	`spl.sh –b stop`	`spl –b stop`
Start Business Application Server	`spl.sh –s start`	`spl –s start`
Stop Web Application Server	`spl.sh –w start`	`spl –w start`

genappvieweritems – generate AppViewer

*Note: The **`splenviron[.sh]`** utility must be executed before this utility can be used. See splenviron – Set Environment variables for details.*

Note: This utility is only executed if AppViewer is used in your environment.

If the environment is used for reference or development then it may be necessary to regenerate the **appViewer** component from the metadata. A utility is provided that runs a number of provided background processes to regenerate the **appViewer** from the current environment.

Command Usage:

Linux/Unix:

```
genappvieweritems.sh [-j] <job> [-Dshv]
```

Windows:

```
genappvieweritems.cmd [-j] <job> [-Dshv]
```

Where:

-h	Show usage.
blank	Execute all extract jobs
-v	Display Version
-j *<job>*	Execute specific <job> from the following list: • **F1-AVALG** - Generate XML file(s) for Algorithm data • **F1-AVMO** - Generate XML file(s) for Maintenance Object data • **F1-AVTBL** - Generate XML file(s) for Table data • **F1-AVTD** - Generate XML file(s) for To Do Types XML • **F1-AVBT** - Generate XML file(s) for Batch Control Types XML
-s	Silent Mode (logs only)
-D	Debug Mode enabled (development use only).

Samples:

```
$ genappvieweritems.sh
…
Application Viewer is delivered with the system including cobol source code
and xml services. This script will extend Application Viewer capabilities
on site by generating additional items.

The Following Programs will be ran
F1-AVALG     Generate XML file(s) for Algorithm data
F1-AVMO   Generate XML file(s) for Maintenance Object data
F1-AVTBL     Generate XML file(s) for Table data
F1-AVTD   Generate XML file(s) for To Do Types XML
F1-AVBT   Generate XML file(s) for Batch Control Types XML

The Application EAR file will also be re-created if required.
Proceed (Y/N)?
…
Calling F1-AVALG
program F1-AVALG got a 0 response code
Calling F1AVMO
program F1-AVMO got a 0 response code
Calling F1-AVTBL
program F1-AVTBL got a 0 response code
Calling F1AVTD
program F1-AVTD got a 0 response code
Calling F1-AVABT
program F1-AVABT got a 0 response code
If you received a non response code 0 above, you should consult the
```

```
logfiles
```

*Note: For platforms that use WAR/EAR files, the **genappvieweritems** utility will automatically rebuild the WAR/EAR files ready for deployment (deployment will need to be performed if **WEB_ISAPPVIEWER** is set to true).*

This generates the HTML files to be included in the appViewer application. This will only generate the necessary files from the current environment. To deploy the appViewer, the relevant option of initialSetup – Maintain Configuration Settings command must be executed to deploy rebuild the WAR file and redeploy the application.

*Note: If an unauthorized user attempts to execute this command the following error message – "**ERROR: Could not create a backup of log file.**" is output.*

initialSetup – Maintain Configuration Settings

*Note: The **initialsetup[.sh]** script replaces the **gen*[.sh]** script provided with previous releases of the Oracle Utilities Application Framework.*

*Note: The **splenviron[.sh]** utility must be executed before this utility can be used. See splenviron – Set Environment variables for details.*

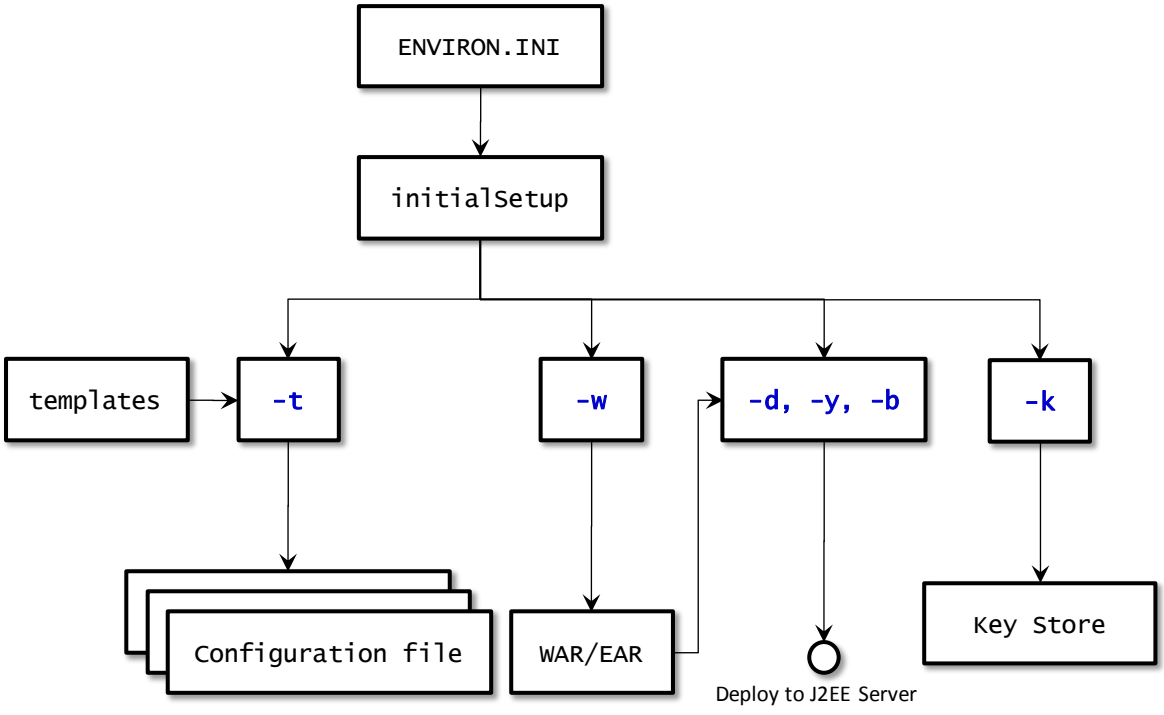
*Warning: This command will reset all configuration files to template settings. Any direct customization to configuration files will be lost. Backup configuration files prior to running this script. If changes are necessary for your site then use user exits or custom templates to retain settings across executions of **initialsetup[.sh]**.*

During the installation and configuration process a number of configuration files used by the components of the architecture are built to be used by the various components of the architecture. The utility takes the ENVIRON.INI settings and using a set of provided templates (located in the etc directory), builds the necessary configuration files for the product components.

This utility has three functions:

- Build/Rebuild the configuration files from templates.
- Build/Rebuild the WAR/EAR files used by the product.
- Deploy the WAR/EAR files to the J2EE Web Application Server (*IBM WebSphere/ND only*). For customers using *native* mode, manual redeployment is necessary.
- Build/Rebuild the keystore files.

This concept is shown in the diagram below:



While this utility is used at installation time and configuration to reflect configuration settings in the product, it can also be used to reset the configuration files to the original settings as well as reflect changes to the ENVIRON.INI - Environment Configuration File.

Command Usage:

Linux/Unix:

```
initialSetup.sh [-h] [-t] [-w] [-d] [-b] [-y] [-v] [-k]
```

Windows:

```
initialSetup.cmd  [-h] [-t] [-w] [-d] [-b] [-y] [-v] [-k]
```

Where:

blank	Process Templates, Build WAR/EAR files and Deploy to J2EE Web Application Server in one process.
-h	Show usage.
-t	Process Templates only
-w	Build WAR/EAR files only
-d	Deploy WAR/EAR files only (IBM WebSphere/ND only)
-b	Deploy Business WAR/EAR files only (IBM WebSphere/ND only)
-y	Deploy Web Application WAR/EAR files only (IBM WebSphere/ND only)
-v	Display Version
-k	Generate Key store

Examples:

```
$ initialSetup.sh
100207.02:37:33 <info> Template generation step.
```

```
100207.02:37:43 <info> FW template generation step.
100207.02:37:43 <info> Create war file for SPLApp.war.
100207.02:38:14 <info> Create war file for XAIApp.war.
100207.02:38:26 <info> Create war file for appViewer.war.
100207.02:39:14 <info> Create war file for help.war.
100207.02:41:11 <info> FINISHED INITIAL INSTALL SETUP at Thu Aug  7
02:41:11 EST 2009
100207.02:41:11 <info> See file
/spl/TRAINING/logs/system/initialSetup.sh.log for details
```

*Note: If an unauthorized user attempts to execute this command the following error message – "**ERROR: Could not create a backup of log file.**" is output.*

iwsdeploy - Inbound Web Services Deployment

*Note: The **splenviron[.sh]** utility must be executed before this utility can be used. See splenviron – Set Environment variables for details.*

The Inbound Web Services deployment utility, **iwsdeploy**, creates the Web Services WAR file and also deploys that WAR file to the servers. The process does the following:

- Extracts the Inbound Web Services that are new or changed as indicated on the deployment record (status of *Needs Repdeployment*).
- For each Web Service generates a number of artefacts required by the Web Application Server to define and execute the Web Service.
- Builds the components into a WAR file called *Webservices.war* located in the **$SPLEBASE/splapp/iws/gen** directory (or **%SPLEBASE%\splapp\iws\gen** directory on Windows).
- Deploys the WAR file to the container configured.

It also is possible to perform this activity using the Inbound Web Services Deployment online function. It is recommended that in non-production both methods are available but in production **iwsdeploy** is used.

The **iwsdeploy** does not have any options at the present time.

Command Usage:

Linux/Unix:

```
iwsdeploy.sh
```

Windows:

```
iwsdeploy.cmd
```

Common Operations

There are a number of common operations that a site will perform on the product. This section outlines the steps involved in these common operations.

Note: The utilities in this section are not to be used by customers using Oracle WebLogic native mode.

Starting an Environment

Note: This section will outline a particular method for starting the product using the supplied utilities. Sites can use the consoles and utilities provided by the Web application server/database vendors to start the product as an alternative.

To ensure a successful startup of the product the components should be started in the following order:

- The database server must be started according to local standards. This includes any communications software such as listeners to enable the product to communicate to the database. After starting the database server, the batch interface can be used.
- The Business Application Server must be started to enable the web application server to use the business objects and the business object conduit to accept web transactions.
- The web application server must be started to enable web clients to access the screens and business objects. After starting the Business Application Server and the web application server, the XAI incoming calls, the batch interface, and online users have access to the system.
- The end users can start the browser to access the product front-end screens.
- Optionally, if the Multi-purpose Listener (MPL) is configured correctly it is also started to support outgoing XAI transactions as well as enable incoming calls from JMS and File.

Starting All Tiers on a Single Server

If the Business Application Server and web application server for an environment are on the same physical machine they can be started using the following set of tasks:

- Start the database using the utilities provided by the database vendor.
- Log on to the server containing the Web application server and/or Business application server using the administration account for the desired environment (for example, **splsys**).
- Execute the following command to attach to the desired environment:

  Linux/Unix:

  ```
  <SPLDIR>/<environment>/bin/splenviron.sh –e <environment>
  ```

Windows:

```
<SPLDIR>\<environment>\bin\splenviron.cmd –e <environment>
```

Where ***<SPLDIR>*** is the mount point defined for software the environment and ***<environment>*** is the name of the environment to start.

- Start the Web application server, Business Application Server and MPL using the following command:

  Linux/Unix:

  ```
  spl.sh start
  ```

  Windows:

  ```
  spl.cmd start
  ```

Refer to the spl – Start/Stop Environment for more options.

Note: As an alternative, it is possible to start the Web application server and business application tiers using the console or utilities provided with the J2EE Web application server software.

The script will display the startup messages as dictated by the J2EE Web application server vendor.

*Note: If an unauthorized user attempts to execute this command the following error message – "**…MUST be writable by admin userid**" is output.*

Starting/Stopping at Boot Time (UNIX/Linux)

One of the implementation questions that may arise is to start all the environments at UNIX/Linux boot time. This is possible by writing a script and placing it in **/etc/init.d** (or equivalent for your platform). A suggested standard is to provide a script that takes a parameter of start or stop. The script could then be used to start or stop product environments on the machine:

```
#!/usr/bin/ksh
#
# Purpose  : Start/Stop all copies of the product on a machine
#

Usage() {
  echo "Usage :"
  echo "  $0 [start|stop]"
  exit 1
}

#-----------------------------------------------------------------
#
#################################################################
#
```

```
# Main

# check command line arguments
if [ "$#" -eq "0" ]
then
 Usage
 exit 1
fi

# Loop through all environments in /etc/cistab

if [ ! -f /etc/cistab ]
then
 echo "/etc/cistab file does not exist. Product is not installed correctly"
 exit 1
fi

cat /etc/cistab | while IFS=: read _env _filler1 _splebase _splapp _filler2
_start
do
  # Only environments with the start parameter set to Y should be started
  if [ ${_start} = "Y" ]
  then
      if [ -d ${_splebase} ]
      then
      # Determine owner of the environment
      export OWNER=`perl ${_splebase}/bin/getconfvalue.plx -k SPLUSER`

      # Format start command

       _startcmd="${_splebase}/bin/splenviron -e {_env} -c ""spl.sh
start"""
       _stopcmd="${_splebase}/bin/splenviron -e {_env} -c ""spl.sh stop"""

       # Run command

       case $1 in
          "start")  su - $OWNER -c "${_startcmd}" ;;
           "stop")  su - $OWNER -c "${_stopcmd}" ;;
                *)  Usage
                    exit 1;;
       esac
     fi
  fi
```

```
done
# Finished
```

Note. The above script is provided as a sample only. Use the above script as an example for any custom scripts to start the product at boot time.

What to Look for in Startup

As outlined in Common Application Logs the application logs all information to application logs during the startup, operation and shutdown of the application. These logs can be used to check that the startup of the product is successful. The logs contain the following sections for a startup (class indicates startup message):

- The Web Application is initialized (class = *web.startup.SPLWebStartup*) within the J2EE Web application server.
- Configuration Settings are loaded from the relevant configuration files (class = *shared.environ.ApplicationProperties*).
- The product is set to Production mode (this denotes Development versus Production settings) (class = *shared.context.ApplicationMode*). Most installations are *Production* mode. Only environments where the Oracle Utilities SDK is used will not be in *Production* mode.
- The state of compression is verified (class = *web.dynamicui.TransformServletHelper*). Refer to Web application server Configuration for details of this setting.
- The framework used by the product is initialized and settings within the framework are prepared to be loaded (*class = support.context.ContextFactory*).
- The metadata is loaded into memory for configuration control (*class = shared.context.ContextLoader*).
- Any checks for any customizations (*class = shared.environ.ContextManagedObjectSet*). In most cases, environments that do not have any product customizations will report a warning about a resource not loading. This can be ignored.
- Any lookups are loaded into memory (*class = support.context.ComponentContainerLookupHelper*). Lookups are metadata used to enumerate valid values for flags, common values etc.
- Additional metadata is loaded into memory (*class = support.context.ContextFactory*). The metadata used to configured the product includes entities, Code Descriptions, algorithms, batch controls, components, Change Handlers and COBOL objects (*if used*).
- Hibernate ORM mappings used by the product are loaded (*class = support.context.ApplicationContext*). The number of mappings will vary between releases and parts of the product that are used.
- The connection pool to the database is initialized according to the configuration settings (*class prefix hibernate.**). If the connection information is incorrect or the

database is down the connection pool connection will retry (according to the configuration settings). If this is the case you will see the connection information and error messages, such as "Connections could not be acquired from the underlying database!" in this log.

Note: The messages seen will vary depending your database type and version.

- A successful database connection is shown in the message "Done building hibernate session" (*class = support.context.ApplicationContext*). A number of additional messages may appear as dictated by the database vendor to indicate versions and connectivity information.
- The database statement cache is initialized within the product (*class = support.sql.PreparedStatementImpl* and *class = support.context.CacheManager*).
- The owner of the system is initialized. This identifies the application owner for implementation purposes. In all cases the implementation value is "CM" for Custom Modification. Other values are supported for Oracle internal use only.
- If COBOL is used for the product then the COBOL Child (or *Worker*) Java Virtual Machines (JVM) are initialized (*class = cobol.host.CobolHostStartup*). During the startup of the JVM's various startup messages will indicate the status of each JVM startup (*class prefix cobol.host*). Each JVM will have individual messages outlining loading and startup of the JVM for COBOL/java integration (JVM number is indicated in the message). Completion of COBOL loading is indicated by message "Remote JVM setup complete" (*class = cobol.host.RemoteJVM*). As COBOL components are detected additional messages will appear in the log to load additional metadata necessary for the execution of the COBOL/java interface (*class prefix support.cobol* and *cobol.mem*).
- The Web application server/Business Application Server static cache is then loaded (*class = api.globalContext.GlobalContextHelper*) which includes:
    - Preloading language settings (*class = web.startup.PreloadLoginInfo*). If preloading is enabled then the progress of preloading is shown on the startup log. Preloading ends with message "XSLT main preload" (*class = web.startup.PreloadLoginInfo*).
    - Loading product based style sheets (XSL) for screen generation.
    - Navigation Keys (for static menus and context sensitive menus) (*class = web.dynamicui.NavigationInfoCache*)
    - Metadata is loaded as indicated (*class = support.context.CacheManager*)
    - Service Interceptors are loaded (*class = api.serviceinterception.InterceptorRepository*)
    - Menus are loaded (*class = domain.web.MenuLoginService*)
    - Navigation information is loaded (*class = domain.web.SystemLoginInfoHelperService*)
    - Service definitions are loaded (*class = service.metainfo.MetaInformationRepository*)

- Installation record defaults are loaded (*class = web.common.WebInstallationDataHelper*)

- If the online batch daemon is enabled then the daemon is loaded into memory and started (*class = grid.node.DistributedGridNode* and prefix *grid.space*). Any work to be detected will result in additional messages (class = *grid.node.WorkProcessor*).
- The Web service adapter (XAI) component is then loaded (delay is configurable) with similar messages as the root application startup. Refer to the top of this list to reference the messages that are loaded.

Once the application is loaded the J2EE Web application server will indicate the product is available (the message for this varies – refer to the J2EE Web application server documentation for details).

Stopping an Environment

Note: This section will outline a particular method for starting the product using the supplied utilities. Sites can use the consoles and utilities provided by the Web application server/Database vendors to start the product as an alternative.

To ensure a successful shut down of the product the components should be stopped in the following order:

- The end users should shut down the browser containing the product front-end screens.
- The MPL must be shutdown (if used) to prevent outgoing XAI transaction from being processed.
- The Web application server must be shutdown to disable web clients' access to the system. After the web application server is shutdown, end users do not have access to the system but batch processes may still run.
- The Business Application Server must be shutdown to disable the Web application server completely.
- The database server must be shut down according to local standards. This includes any communications software such as listeners to enable the product to communicate to the database. At this point all users (batch and online) do not have access to the environment.

Stopping All Tiers on a Single Server

If the Business Application Server and web application server for an environment are on the same physical machine they can be stopped/shutdown using the following set of tasks:

- Log on to the server containing the Web application server and/or Business application server using the administration account for the desired environment (for example, **splsys**).
- Execute the following command to attach to the desired environment:

Linux/Unix:

```
<SPLDIR>/<environment>/bin/splenviron.sh -e <environment>
```

Windows:

```
<SPLDIR>\<environment>\bin\splenviron.cmd -e <environment>
```

Where ***<SPLDIR>*** is the mount point defined for software the environment and ***<environment>*** is the name of the environment to stop.

- Stop the Web application server, Business Application Server and MPL using the following command:

  Linux/Unix:

  ```
  spl.sh stop
  ```

  Windows:

  ```
  spl.cmd stop
  ```

  Refer to the spl[.sh] utility for more options.

  Note: As an alternative, it is possible to stop the Web application server and business application tiers using the console or utilities provided with the J2EE Web application server software.

  The script will display the shutdown messages as dictated by the J2EE Web application Server vendor.

- Stop the database using the utilities provided by the database vendor.

What to Look For in Shutdown Messages

As outlined in Common Application Logs the application logs all information to application logs during the startup, operation and shutdown of the application. These logs can be used to check that the shutdown of the product is successful. The logs contain the following sections for a shutdown (class indicates message class used):

- If the online batch daemon was enabled, it is shutdown (*classes = grid.node.OnlineGridNode, grid.node.DistributedGridNode, grid.space.SpaceManager, grid.space.TaskScheduler, grid.space.TaskScheduler* and *grid.space.ThreadPool*). The "Thread pool shutting down" message indicates a successful shutdown.
- The Web application server/Business Application Server applications are asked to shutdown (*class = web.startup.SPLWebStartup*).
  - JMX connectors to the product are shutdown
  - The Application Context within the J2EE Web application server is shutdown. This may be delayed if COBOL is installed.
- If COBOL is used, then the COBOL Child (or Worker) JVMS are shutdown. The term used is *shunned*. Each JVM is shunned individually.

Note: A message "java.net.SocketException closing connection" may be displayed. This indicates that the socket has been closed.

- Database connections are closed (*class = hibernate.impl.SessionFactoryImpl*).
- Application shutdown is complete when the message "(*web.startup.SPLWebStartup*) Application Context shutdown successfully" is displayed.

Monitoring

This section outlines some basic monitoring regimes and methods for the product. It is highly recommended that you read the Oracle Utilities Application Framework Performance Troubleshooting Guides KB Id: **560382.1** on My Oracle Support.

During monitoring you are typically looking for unusual activity and seeing if the current configuration of the product can handle the peaks and troughs of usage.

Unusual activity is activity that is not representative of the normal activity. For example, maybe during a marketing campaign the call center traffic doubles. This would be regarded *unusual activity*. At this point the current configuration may not be configured to handle the traffic so the problem needs to be identified and the configuration changed to cater for the new load.

Also during normal operations underlying problems may surface in the form of long running transactions, increases in error rates (in logs and timeouts) or *runaway transactions*. *Runaway transactions* are transactions that seem to be looping. These can be caused by data inconsistencies or bugs. Most of them are due to an unusual combination of data entries.

Some customers collect usage information to identity and analyze unusual activity. This is known as Site Profiling, Capacity Planning or Availability Planning. This is typically *Proactive* activity.

The product stores usage information within the database that can be extracted for this purpose. This section outlines the methods and techniques you can use to extract this information reactively and proactively.

Monitoring Regimes

Typically the art of monitoring is the collection and analysis of various pieces of information and then making changes to the configuration to address any issues or problems that occur.

With the various monitoring facilities available in the product a combination that is valid for the site becomes a monitoring regime for that site. Typically, monitoring regimes pick up trends in the business or traffic volumes that require changes to the configuration. As part of the implementation of the product the monitoring regime for your site should be determined.

Typically the monitoring regimes that are chosen fall into a number of categories:

- **Reactive** - Monitoring for any exception after it happens and making changes to the configuration to prevent the exception from occurring again. This is the most common regime adopted by IT groups. The only problem with this approach is that you have to experience potentially threatening outages before stabilization happens.
- **Proactive** - Setting monitoring tolerances so that exception conditions are recognized before they happen and making configuration changes to prevent them from happening. This is also known as *Problem Anticipation* or *Problem Prevention*. This is the goal of most of the IT groups to ensure high availability.

- **Mixed** - This is a mixture of pro-active and re-active regime. This is not uncommon.

Monitoring Client Machines

The product's front end is the Microsoft Internet Explorer browser. Typically any Internet Explorer or operating system monitoring specified by Microsoft can be performed against the client to yield performance information.

While collecting this information can be performed using various tools, it is usually not applicable in all monitoring situations unless the client machine is below the specification outlined in the Installation Guide for the platform and version of the product you are using. The browser collection points specified here are typically the ones that are more applicable to the product than all of the available ones for the client.

Refer to the Microsoft documentation on how to fully monitor a client machine for performance information

Monitoring The Desktop

One of the areas that customers tend to monitor is the desktop client. Typically this involves using tools provided by Microsoft (and other vendors) to collect typical statistics, such as cpu, disk activity, memory usage and network usage. It is possible to monitor the client using the following tools:

- **Desktop vendor tools** (Performance Monitor) – The Performance Monitor (located in the "Administration Tools" menu from Windows) is a starting point for monitoring the client. Refer to Microsoft documentation on what aspects of a client machine to monitor.
- **Network Monitor** (*netMon* or other) – Windows Server includes a network capture facility that is handy to locate problems on a client machine. Alternatives are available such as Ethereal etc.
- **Network Latency** - Network tools like *ping* and *traceroute* measure latency by determining the time it takes a given network packet to travel from source to destination and back, the so-called round-trip time. Round-trip time is not the only way to specify latency, but it is the most common. Inconsistent ping times or long ping times can indicate network issues.
- **Bandwidth Saturation levels** - A number of tools exist for computer networkers to measure the bandwidth of network connections. On LANs, these tools include *netperf* and *ttcp*.
- **Packet Loss** - Packet loss is when data packets appear to be transmitted correctly at one end of a connection, but never arrive at the other. This might be because:
    - Network conditions are poor and the packet became damaged in transit.
    - The packet was deliberately dropped at a router because of congestion.
- Packet loss can be detected from the client PC using *netstat* and calculating the percentage of the *Segments Sent* that become *Segments Retransmitted*.

Note: ping and traceroute also include packet loss statistics.

- **Failed Connection Attempts** - When the client and/or server cannot accept a connection it generates a *Failed Connection Attempt* on either the client or the server (or both). A large number of *Failed Connection Attempts* can indicate networking or capacity issues on the client or server. The most common cause is that the accept queue on the network parameters (usually on the network cards) is full, and there are come requests waiting on the sync queue (usually on the network card).

Client Debug facility

Before a problem is to be registered with Oracle support, the transaction that caused the problem should be traced to help support solve the issue quickly. A debug facility is provided within the product to help capture this additional information.

Logging of debug information can be set at a global level or at a *local* level. The global debug setting is not recommended for a production system as it reduces overall performance and therefore is not covered in this document.

The *local* level enables you to navigate to the problem area and then to switch debugging on for that individual user to recreate the problem. You can then collate the debug information to be sent to support.

To use this facility you must specify an additional parameter at the end of the URL. For example:

`http://<host>:<port>/<server>/cis.jsp?debug=true`

Where:

<host>	Web Application Server hostname
<port>	Port allocated to product installation
<server>	Context for the product at installation time

Note: For the user to have debug access their userid must have "Change" access to service ***F1DEBUG****.*

After the debug control menu is displayed, you navigate to the screen where the problem is encountered and then enable *Global Debug* by *toggling* the checkbox on. To turn off *Global Debug*, *toggle* the check box off. It is recommended to select *Trace All* for effective tracing. The other options are used by Developers only. The trace information is written to the **spl*.log** in the **`$SPLSYSTEMLOGS`** (**`%SPLSYSTEMLOGS%`** in Windows).

Note: The product uses ***spl_web.log*** *and* ***spl_service.log*** *but* ***spl_service.log*** *or may not appear depending on the installation type, therefore the name* ***spl*.log*** *is used.*

Debug allows specific information to be logged:

- **Client Data** – Data presented to the browser. This pops up an additional window displaying the object as it is built.
- **Server Data** – Data presented to the server. This pops up an additional window displaying the object as it is received by the server.
- **Trace time** – Include time tracing in the log.

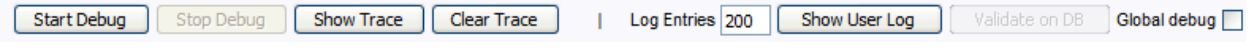
- **COBOL buffers** (if COBOL is used), Debug List Info, Debug Filter and Grid Display Time – Used for development to display internal information and filter for specific information. It is recommended that these options should not be used unless performing development.
- **Trace All** – Enable all trace modes below except Trace SQL Parameters.
- **Trace Output** – Dump output from all calls
- **Trace SQL** – Dump SQL statements
- **Trace SQL Parameters** – Dump all result sets (Warning: This is not recommended for production systems as it will result in performance degradation.)
- **Program Start** – Write a record for ever module start
- **Program End** – Write a record for ever module end

Most tracing in non-development uses *Trace All* unless otherwise instructed by Oracle Support. All debug information is written to the **spl*.log** files.

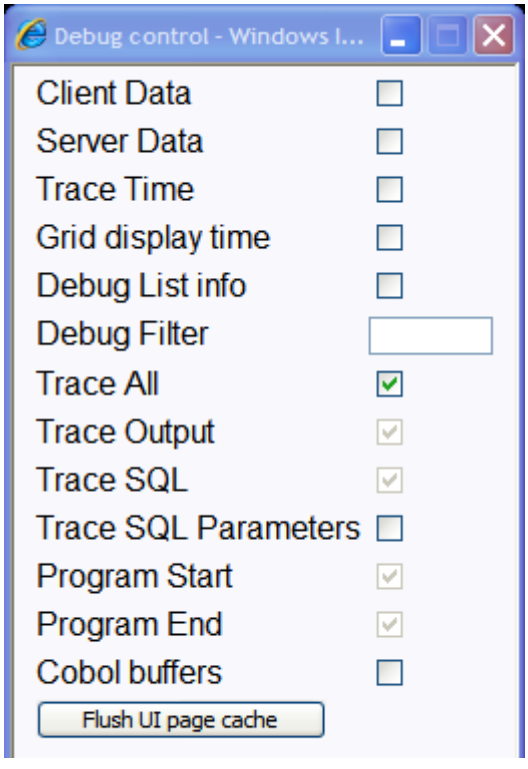
Steps to using the debug facility

To use the debug facility you follow the process:

- Add `?debug=true` to your URL for the product. This will display the debug buttons on the browser screen as shown below:



  - **Start Debug** – Start the logging of the transaction.
  - **Stop Debug** – Stop the logging process
  - **Show Trace** – Show trace information (Configuration based objects only)
  - **Clear Trace** – Clear Trace Information
  - **Show User Log** – Show debug information for the user (line limit is configurable).
  - **Validate on DB** – SDK Use only
  - **Global Debug** – Set debug level.
- Select *Global Debug* to specify the level of debug information. This will display the *Debug Control* window where you should ensure that *Trace All* is selected. Other options should only be used if instructed by Oracle Support. A sample of the Debug Control dialog is shown below:



- Navigate to the transaction that you wish to trace as a user would normally operate. Press "Start Debug" to initiate debug.
- Run the transaction that you want to trace and to recreate the issue. While you work the trace information is written to the log files.
- Deselect *Global Debug* or press *Stop Debug* so that debugging is disabled. This will stop debug code writing to the writing to the log. If you select *Show User Log* the log lines output by the debug facility are displayed (*up to the line limit specified*). This will only show lines applicable to the Current User only.

Note: If the userid is shared across multiple physical users then the information may contain debug information from multiple sessions.

Monitoring Web/business Application Server

There are a number of methods that are available for monitoring a Web Application from a J2EE Web application server:

- **Java Management Extensions (JMX)** – Most Web application servers expose JMX Management Beans (MBeans) to allow JMX browsers to view and use this information. Java 6 has a predefined set of MBeans that can be enabled automatically.
- **Web application server console** – All Web Application Servers offer a web based console that provides both administration and basic monitoring functions. These are usually sufficient for spot real time checking of tolerances and basic monitoring. Some console use calls to JMX API's provided by the Web application server vendor and built into Java 6 (and above).
- **Command Based Utilities** – Apart from the console, most Web application server

vendors offer a command line utility to extract performance information (or perform administration). Most console utilities call JMX MBeans and provide a command line interface into JMX that can be used natively.

- **Log-based monitoring** – Most Web application servers provide standardized logs that can be analyzed using consoles, log monitors or simple scripts.
- **Native OS utilities** – Most operating systems are becoming java aware and provide OS and Java monitoring from OS monitoring facilities.

Refer to the Oracle Utilities Application Framework Performance Troubleshooting Guides KB Id: **560382.1** on My Oracle Support for details of monitoring aspects of the product.

JMX Based Monitoring

With the advent of Java Management Extensions (JMX) technology into base java, it is possible to use the technology to monitor and manage java infrastructure from a JSR160 compliant JMX compliant console (or JMX browser). Whilst the J2EE components of the product can use basic JMX statistics such as Memory usage, Threads, Class information and VM summary information, there are application specific JMX classes added to the product to allow greater levels of information to be display and additional operations.

The Oracle Utilities Application Framework has implemented a set of product specific JMX classes on the Web Application Server and Business Application Server tiers of the architecture to allow the following:

- Management of the cache of the Web Application Server. See Server Cache Management for more details of this cache.
- Collection of JVM information and performance statistics for memory, thread usage and operating system level information. Most of these are extensions of java.lang.management classes.
- Collection of service based performance information for SLA tracking on the Business Application Server.

To use this facility the facility must be configured and enabled to allow the collection of the relevant information. This can be done at installation time by using the following configuration settings:

Configuration Setting	Deployment details
`WEB_JMX_RMI_PORT_PERFORMANCE`	Port Number used for JMX based management for Web Application Server.

`ouaf.jmx.splwls`
`g.base.support.`
`management.mbean.JVMInfo`
	Globally enable or disable `JVMInfo` Mbean (setting in spl.properties). Default is **enabled**.

`ouaf.jmx.com.splwg.base.web.mbeans` `.FlushBean`	Globally enable or disable `FlushBean` Mbean (setting in spl.properties). Default is **enabled**.
`BSN_JMX_RMI_PORT_PERFORMANCE`	Port Number used for JMX based

Configuration Setting	Deployment details
	management for Business Application Server.
ouaf.jmx.com.splwg.ejb.service.management.PerformanceStatistics	Globally enable or disable **PerformanceStatistics** Mbean (setting in spl.properties). Default is **enabled**
BSN_JMX_SYSUSER	Default JMX Userid for both Web Application Server and Business Application Server
BSN_JMX_SYSPASS	Default JMX Password for both Web Application Server and Business Application Server

These settings are registered in the ENVIRON.INI for setting in the relevant configuration files. It is important that the values used for these port numbers are unique across all environments within a particular machine. The security used for these ports are defined as outlined in the JMX Security section of this document.

Web Application Server JMX Reference

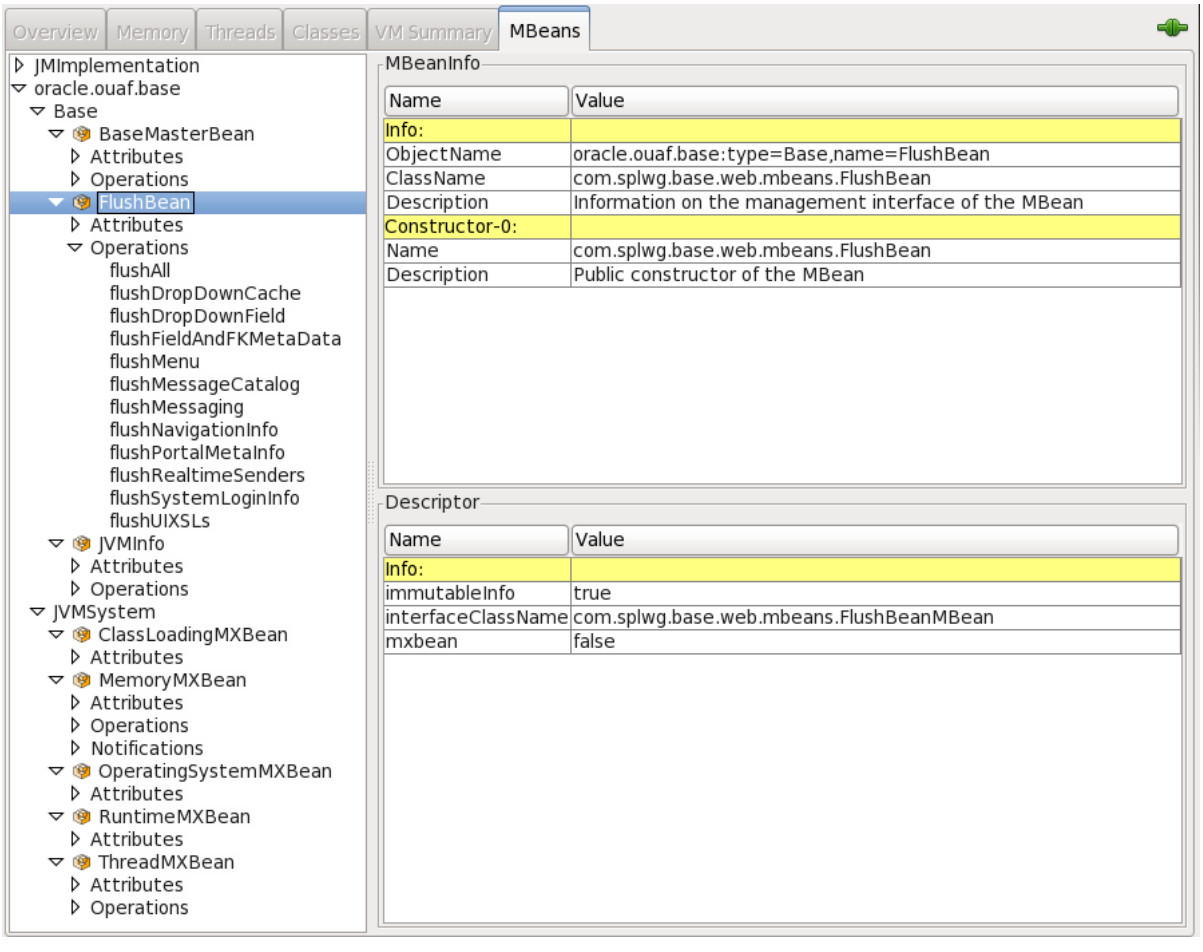
Once configured a JMX client (e.g. **jconsole**) can be used to connect to the JMX information using the following Remote Connection string:

`service:jmx:rmi:///jndi/rmi://<host>:<jmx_port>/oracle/ouaf/webAppConnector`

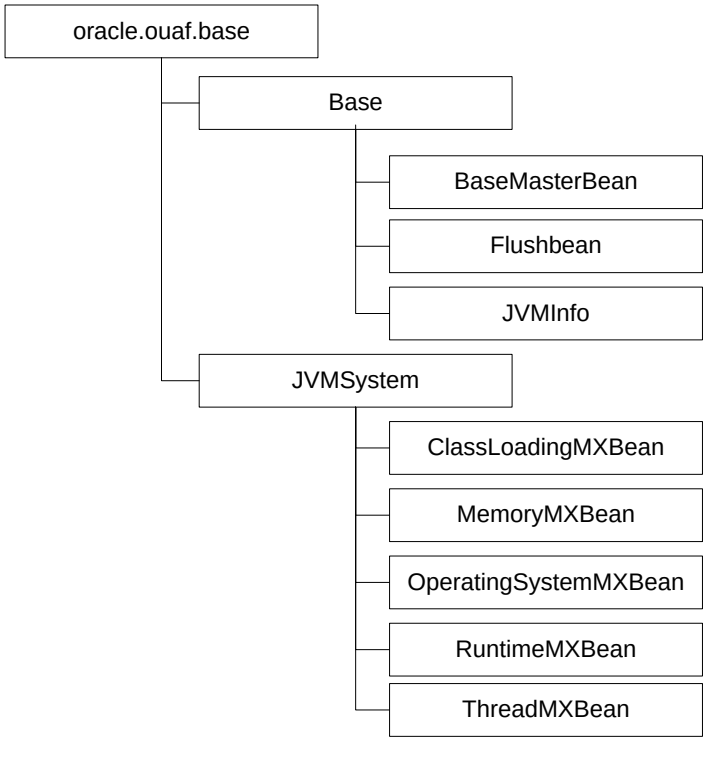
Where:

<host>	The Web Application Server host name
<jmx_port>	The JMX Port specified using **WEB_JMX_RMI_PORT_PERFORMANCE** from the ENVIRON.INI configuration file.

The credentials provided to the JMX console are as configured in JMX Security. Upon successful connection to the JMX port and host with the correct credentials provides access to the Mbean information. The figure below illustrates the successful connection to the JMX Mbeans using **jconsole** (*as an example*):



The structure of the Mbean is shown by the figure below:



The following table summarizes the Mbean attributes and operations for the Web Application Server:

Mbean	Arguments	Usage
BaseMasterBean Attributes		
NumberOfMbeans	None	Returns number of active Mbeans
MBeanList	None	Returns an array with the list of Mbeans defined to this Master Mbean
BaseMasterBean Operations		
disableMbean	Mbean Name	Disables Mbean with designated name
enableMbean	Mbean Name	Enables Mbean with designated name
disableMbean	None	Returns a list with the list of Mbeans defined to this BaseMasterBean. Names can be programmatically used to supply parameters to **disableMbean** and **enableMbean**.
enableJVMSystemBeans	None	Enables base JVM system Mbeans
disableJVMSystemBeans	None	disables base JVM system Mbeans
FlushBean Attributes		
VersionInfo	None	Returns string of base version number of Flush Mbean.
CompleteClassPath	None	Returns classpath name of Flushbean Mbean.
FlushBean Operations (Refer to Server Cache for details of this cache).		
flushAll	None	Reset all elements in online data cache
flushDropDownCache	None	Resets cached elements of the online drop down lists in online data cache
flushDropDownField	None	Resets drop down fields in online data cache (*Development use only*)
flushFieldAndFKMetaData	None	Resets Field and Foreign Key Meta Data in online data cache
flushMenu	None	Reset Menu items in online data cache
flushMessageCatalog	None	Reset field labels in online data cache
flushMessaging	None	Reset messages in online data cache
flushNavigationInfo	None	Reset navigation information in

Mbean	Arguments	Usage
		online data cache
flushPortalMetaInfo	None	Reset portal and zone information in online data cache
flushSystemLoginInfo		Reset security information in online data cache
flushUIXSLs	None	Reset user interface style sheets in online data cache
JVMInfo Attributes		
CompleteClassPath	None	Displays the class path of the JVMInfo mbean
JVMInfo Operations		
classPath	None	Returns the full classpath used by the online JVM
systemSettings	None	Returns the attributes of the JVM for debugging and support purposes.
ClassLoadingMXBean Attributes		
LoadedClassCount	None	Returns the number of classes that are currently loaded in the JVM
TotalLoadedClassCount	None	Returns the total number of classes that have been loaded since the JVM was last started.
UnloadedClassCount	None	Returns the total number of classes unloaded since the Java virtual machine has started execution.
Verbose	None	Enables or disables the verbose output for the class loading system. Default is **false** (*disabled*)
MemoryMXBean Attributes		
HeapMemoryUsage	None	Returns the current memory usage of the heap that is used for object allocation. Initial, Committed, Maximum and Used memory statistics are provided for Heap memory
NonHeapMemoryUsage	None	Returns the current memory usage of non-heap memory that is used by the JVM. Initial, Committed, Maximum and Used memory statistics are

Mbean	Arguments	Usage
		provided for Non-Heap memory
ObjectPendingFinalization	None	Returns the approximate number of objects for which finalization is pending (used for diagnosing memory leaks).
Verbose	None	Enables or disables the verbose output for the memory system. Default is **false** (*disabled*)
MemoryMXBean Operations		
gc	None	Initiate garbage collection
MemoryMXBean Notifications		
javax.management.Notification	None	Used for low memory notifications. Notification Types supported: (java.management.memory.threshold .exceeded, java.management.memory .collection.threshold.exceeded)
OperatingSystemMXBean		
MaxFileDescriptorCount	None	Returns the File Descriptor Maximum Limit in force on the JVM
OpenFileDescriptorCount	None	Returns the number of Open File Descriptors currently used by JVM
CommittedVirtualMemorySize	None	Returns the amount of committed virtual memory (that is, the amount of virtual memory guaranteed to be available to the running process).
FreePhysicalmemorySize	None	Returns the total amount of free physical memory
FreeSwapSpaceSize	None	Returns the total amount of free swap space
ProcessCpuTime	None	Returns the amount of process CPU time consumed by the JVM
TotalPhysicalMemorySize	None	Returns the total amount of physical memory
TotalSwapSpaceSize	None	Returns the total amount of swap space
Name	None	Returns the operating system name

Mbean	Arguments	Usage
Version	None	Returns the version of the operating system
Arch	None	Returns the operating system architecture
AvailableProcessors	None	Returns the number of available processors to the JVM
SystemLoadAverage	None	Returns the system load average for the last minute.
RuntimeMXBean Attributes		
Name	None	Returns the name representing the running JVM. The returned name string can be any arbitrary string and a JVM implementation can choose to embed platform-specific useful information in the returned name string. Each running virtual machine could have a different name.
ClassPath	None	Returns the Java class path that is used by the system class loader to search for class files.
Starttime	None	Returns the start time of the Java virtual machine in milliseconds. This method returns the approximate time when the JVM started.
ManagementSpecVersion	None	Returns the version of the specification for the management interface implemented by the running JVM
VmName	None	Returns the Java virtual machine implementation name
VmVendor	None	Returns the Java virtual machine implementation vendor
VmVersion	None	Returns the Java virtual machine implementation version
SpecName	None	Returns the Java virtual machine specification name
SpecVendor	None	Returns the Java virtual machine specification vendor
SpecVersion	None	Returns the Java virtual machine

Mbean	Arguments	Usage
		specification version
LibraryPath	None	Returns the Java library path
BootClassPath	None	Returns the boot class path that is used by the bootstrap class loader to search for class files
Uptime	None	Returns the uptime of the Java virtual machine in milliseconds
BootClassPathSupported	None	Tests if the JVM supports the boot class path mechanism used by the bootstrap class loader to search for class files. Returns *false* if not supported; *true* if supported
InputArguments	None	Returns the input arguments passed to the JVM which does not include the arguments to the main method. This method returns an empty list if there is no input argument to the JVM. Typically, not all command-line options to the '`java`' command are passed to the Java virtual machine. Thus, the returned input arguments may not include all command-line options
SystemProperties	None	Returns a map of names and values of all system properties
ThreadMXBean Attributes		
ThreadCount	None	Returns the current number of live threads including both daemon and non-daemon threads
PeakThreadCount	None	Returns the peak live thread count since the JVM started or peak was reset
TotalStartedThreadCount	None	Returns the total number of threads created and also started since the JVM started
DaemonThreadCount	None	Returns the current number of live daemon threads
ThreadContentionMonitoringSupported	None	Tests if the JVM supports thread contention monitoring. Returns *false* if not supported; *true* if supported

Mbean	Arguments	Usage
ThreadContentionMonitoringEnabled	None	Enables or disables thread contention monitoring. Set to *false* to disable; *true* to enable.
CurrentThreadCpuTime	None	Returns the total CPU time for the current thread in nanoseconds. The returned value is of nanoseconds precision but not necessarily nanoseconds accuracy. If the implementation distinguishes between user mode time and system mode time, the returned CPU time is the amount of time that the current thread has executed in user mode or system mode
CurrentThreadUserTime	None	Returns the CPU time that the current thread has executed in user mode in nanoseconds. The returned value is of nanoseconds precision but not necessarily nanoseconds accuracy.
ThreadCpuTimeSupported	None	Tests if the JVM supports CPU time measurement for the current thread. Returns *false* if not supported; *true* if supported
ThreadCpuTimeEnabled	None	Enables or disables thread CPU time measurement. The default is platform dependent. Set to *false* to disable; *true* to enable.
CurrentThreadCpuTimeSupported	None	Tests if the Java virtual machine supports CPU time measurement for the current thread. Returns *false* if not supported; *true* if supported
ObjectMonitorUsageSupported	None	Tests if the Java virtual machine supports monitoring of object monitor usage. Returns *false* if not supported; *true* if supported
SynchronizerUsageSupported	None	Tests if the JVM supports monitoring of ownable synchronizer usage. Returns *false* if not supported; *true* if supported
AllThreadIds	None	Returns all live thread IDs. Some

Mbean	Arguments	Usage
		threads included in the returned array may have been terminated when this method returns
ThreadMXBean Operations		
dumpAllThreads	Locked Monitors, Locked Synchronizers	Returns the thread info for all live threads with stack trace and synchronization information. Some threads included in the returned array may have been terminated when this method returns • Locked Monitors - if *true*, dump all locked monitors • Locked Synchronizers - if *true*, dump all locked ownable synchronizers
findDeadlockedThreads	None	Finds cycles of threads that are in deadlock waiting to acquire object monitors or ownable synchronizers. Threads are deadlocked in a cycle waiting for a lock of these two types if each thread owns one lock while trying to acquire another lock already held by another thread in the cycle
getThreadCpuTime	Thread Id	Returns the total CPU time for a thread of the specified ID in nanoseconds. The returned value is of nanoseconds precision but not necessarily nanoseconds accuracy. If the implementation distinguishes between user mode time and system mode time, the returned CPU time is the amount of time that the thread has executed in user mode or system mode
getThreadInfo	Thread Id	Returns the thread info for a thread of the specified id with no stack trace.
getThreadInfo	Array of Thread Ids	Returns the thread info for each thread whose ID is in the input array ids with no stack trace.

Mbean	Arguments	Usage
getThreadInfo	Thread Id, maxDepth	Returns thread information for a thread of the specified id, with stack trace of a specified number of stack trace elements. The *maxDepth* parameter indicates the maximum number of *StackTraceElements* to be retrieved from the stack trace. This method does not obtain the locked monitors and locked synchronizers of the thread
getThreadInfo	Array of Thread Ids, maxDepth	Returns the thread information for each thread whose ID is in the input array ids, with stack trace of a specified number of stack trace elements. The *maxDepth* parameter indicates the maximum number of *StackTraceElements* to be retrieved from the stack trace. This method does not obtain the locked monitors and locked synchronizers of the threads
getThreadInfo	Array of Thread Ids, locked Monitors, locked Synchronizors	Returns the thread info for each thread whose ID is in the input array ids, with stack trace and synchronization information. This operation obtains a snapshot of the thread information for each thread including: • the entire stack trace, • the object monitors currently locked by the thread if *lockedMonitors* is true, and • the ownable synchronizers currently locked by the thread if *lockedSynchronizers* is true
getThreadUserTime	Thread Id	Returns the CPU time that a thread of the specified ID has executed in user mode in nanoseconds
resetPeakThreadCount	None	Resets the peak thread count to the current number of live threads

Business Application Server JMX Reference

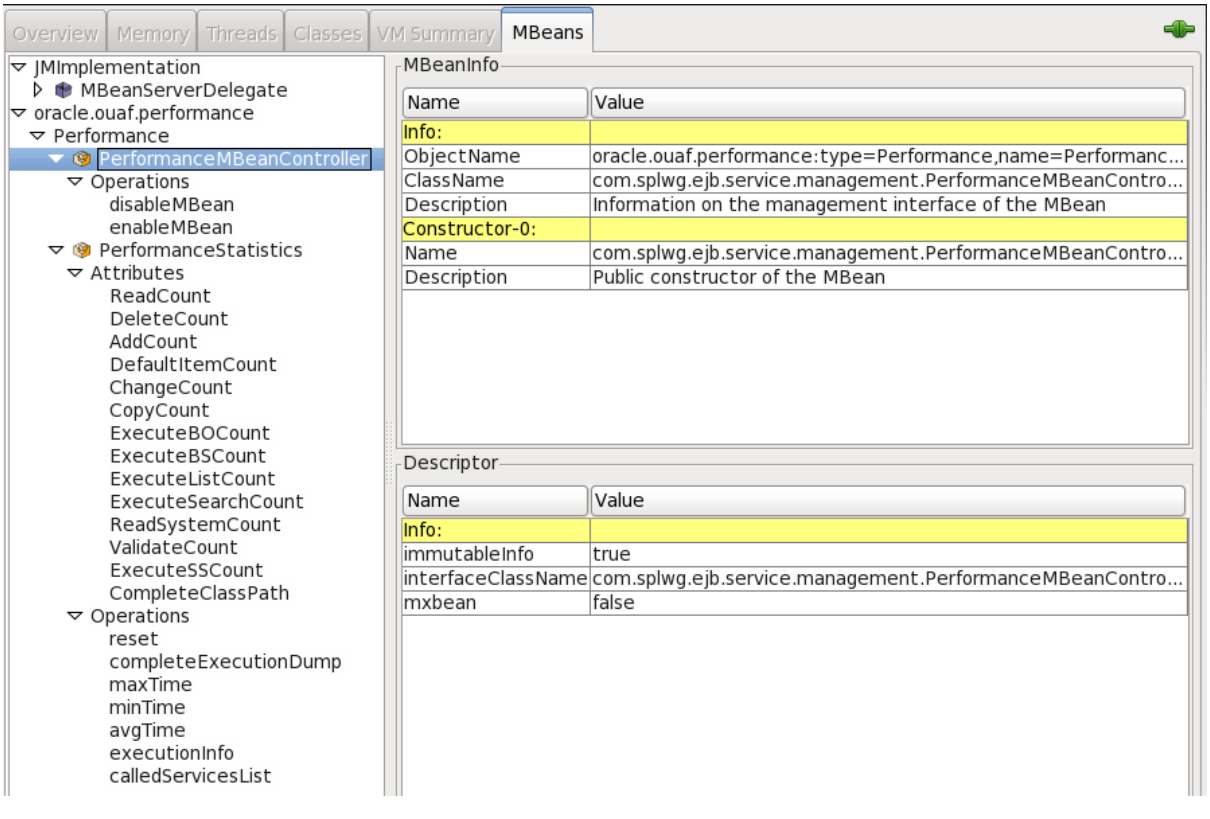
Once configured a JMX client (e.g. **jconsole**) can be used to connect to the JMX information for the Business Application Server using the following Remote Connection string:

`service:jmx:rmi:///jndi/rmi://<host>:<jmx_port>/oracle/ouaf/ejbAppConnector`

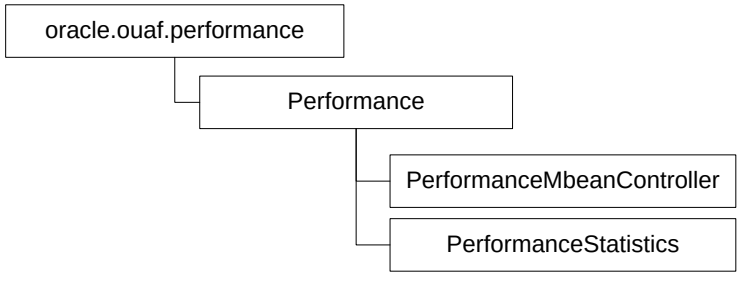
Where:

<host>	The Business Application Server host name
<jmx_port>	The JMX Port specified using **BSN_JMX_RMI_PORT_PERFORMANCE** from the ENVIRON.INI configuration file.

The credentials provided to the JMX console are as configured in JMX Security. Upon successful connection to the JMX port and host with the correct credentials provides access to the Mbean information. The figure below illustrates the successful connection to the JMX Mbeans using **jconsole** (*as an example*):



The structure of the Mbean is shown by the figure below:



The following table outlines the Mbean attributes and operations for the Business Application Server:

Mbean	Arguments	Usage
PerformanceMBeanController Operations		
disableMbean	None	Disable PerformanceStatistics Mbean
enableMbean	None	Enable PerformanceStatistics Mbean
PerformanceStatistics Attributes		
ReadCount	None	Returns number of executed *read object* calls since last reset or last time collection enabled
DeleteCount	None	Returns number of executed *delete object* calls since last reset or last time collection enabled
ChangeCount	None	Returns number of executed *change object* calls since last reset or last time collection enabled
AddCount	None	Returns number of executed *add object* calls since last reset or last time collection enabled
DefaultItemCount	None	Returns number of executed calls to *default the object values* since last reset or last time collection enabled
ExecuteBOCount	None	Returns number of calls to *Business Objects* since last reset or last time collection enabled
ExecuteBSCount	None	Returns number of calls to *Business Services* since last reset or last time collection enabled
ExecuteListCount	None	Returns number of calls to *List based services* since last reset or last time collection enabled
ExecuteSearchCount	None	Returns number of calls to *search based services* since last reset or last time collection enabled
ReadSystemCount	None	Returns number of calls to *Oracle Utilities Application Framework system Objects* since last reset or last time collection enabled
ValidateCount	None	Returns number of calls to *Validate objects* since last reset or last time collection enabled
ExecuteSSCount	None	Returns number of calls to *Service Scripts* since last reset or last time collection enabled
CompleteClassPath	None	Returns the class path used for the Mbeans
PerformanceStatistics Operations		
reset	None	Resets statistical values. See Resetting Statistics for more advice on this operation.
maxTime	Service Name	Returns maximum (worst case) time, in ms, for the designated service since the last reset or last

Mbean	Arguments	Usage
		time collection enabled.
`minTime`	Service Name	Returns minimum (best case) time, in ms, for the designated service since the last reset or last time collection enabled.
`completeExecutionDump`	None	Returns complete statistics for all services executed since the last reset or last time collection enabled. See Execution Dump section for details of format.
`avgTime`	Service Name	Returns average time, in ms, for the designated service since the last reset or last time collection enabled.
`executionInfo`	Service Name	Returns complete statistics for the designated service executed since the last reset or last time collection enabled. See Execution Dump Format section for details of format.
`calledServices`	None	Returns list of services and service types since the last reset or last time collection enabled. See Service Lists for details of format.

Note: The times quoted in the statistics only record times experienced from the Business Application Server down to the data and back. They do not include network time to the Web Application Server, any time spent by the Web Application Server, network time to the browser client or browser rendering times. The Business Application Server time represents the typical majority of the time spent in a transaction.

Web Services JMX Reference

Note: This facility is only available when using the Native Web Services capability as an alternative to the XML Application Integration (XAI) capability.

Once configured a JMX client (e.g. **`jconsole`**) can be used to connect to the JMX information for the Web Services using the following Remote Connection string:

```
service:jmx:rmi:///jndi/rmi://localhost:6590/oracle/ouaf/iwsConnector
```

Where:

<host>	The Business Application Server host name
<jmx_port>	The JMX Port specified in the **`spl.properties`** file (**`spl.runtime.rmi.port`**).

The credentials provided to the JMX console are as configured in JMX Security. Upon successful connection to the JMX port and host with the correct credentials provides access to the Mbean information. The figure below illustrates the successful connection to the JMX Mbeans using **`jconsole`** (*as an example*):

<insert jconsole>

The structure of the Mbean is shown by the figure below:

<insert diagram>

The following table outlines the Mbean attributes and operations for the Web Services:

Mbean	Arguments	Usage
PerformanceMBeanController Operations		
`disableMbean`	None	Disable PerformanceStatistics Mbean
`enableMbean`	None	Enable PerformanceStatistics Mbean
PerformanceStatistics Attributes		
`ReadCount`	None	Returns number of executed *read object* calls since last reset or last time collection enabled
`calledServices`	None	Returns list of services and service types since the last reset or last time collection enabled. See Service Lists for details of format.

JMX Security

By default, when JMX is enabled for either the Web Application Server, Web Service and Business Application Server then a default JMX configuration using simple security is implemented as outlined in http://java.sun.com/javase/6/docs/technotes/guides/management/agent.html.

The simple security system consists of two files that control the access permissions and passwords specified by default for the installation:

Configuration Setting	Location of file	Template
Password File	`scripts/ouaf.jmx.password.file`	`ouaf.jmx.password.file.template`
Access Control File	`scripts/ouaf.jmx.access.file`	`ouaf.jmx.access.file.template`

These files are built by the initialSetup utility using the templates indicated. Refer to the templates or generated files for valid values. The format of these files is dictated by http://java.sun.com/javase/6/docs/technotes/guides/management/agent.html#gdeup.

Note: By default, the passwords stored in these files are in encrypted text. Alternative security schemes are allows as documented in the link above. This will require a custom templates and changes to specific files to implement.

Extending JMX Security

Whilst the base installation of the product uses the basic level of security there are ways of extending the current security:

- If the default security scheme is sufficient for your needs then additional users may be manually added using the user exits for the above files.
- For production it is recommended to implement an SSL based solution as outlined in http://docs.oracle.com/javase/6/docs/technotes/guides/management/agent.html.

Refer to the product Security Guide for more schemes available for this process.

Execution Dump Format

In previous versions (V1.x) of the Oracle Utilities Application Framework based products, it was possible to extract performance information from the Business Application Server using a logging based method using the Oracle Tuxedo **`txrpt`** utility. This facility was useful in tracking performance of individual services over time to detect non-compliance against Service Level Agreement targets. With the advent of later versions of the Oracle Utilities Application Framework, the need for Oracle Tuxedo was removed but there was a need for performance information to be collated.

In the latest version of the Oracle Utilities Application Framework, it is possible to track performance information using JMX to process externally to check performance and check compliance against Service Level Agreements.

To extract the information from the product the following needs to be done:

- Use a JMX browser (or JMX console) product to connect to the Business Application Server JMX port using the appropriate credentials.
- Invoke the `completeExecutionDump` operation from the `PerformanceStatistics` Mbean. This is will return a Comma separated values, with field names in the header record, containing the performance data which can be transferred to the clipboard (or whatever format supported by the JMX client). The format of the CSV is shown in the table below:

Column	Comment
ServiceName	Name of Service
ServiceType	Type Of Service or Action (see Service Lists for valid values)
MinTime	Minimum Service Time, in ms, since last reset
MaxTime	Minimum Service Time, in ms, since last reset
Avg Time	Average Service Time, in ms, since last reset
# of Calls	Number of Calls to Service since last reset
Latest Time	The service time of the latest call, in ms
Latest Date	The date of the latest service call (in format: *YYYY-MM-DD::hh-mm-ss-sss*)
Latest User	The userid of the user who issued the latest call

- (Optionally) Invoke the `reset` operation from the `PerformanceStatistics` Mbean to reset the statistics for the next collection period. Refer to Resetting Statistics for a discussion of this task.

This information can then be post processed in an appropriate analysis tool to determine appropriate actions.

Note: The statistics are active as long the Mbean is enabled or the system is active. Shutting down the Business Application Server with collection of the data may cause data loss for the statistics.

Service Lists

The JMX Performance Mbeans collect information about application services that have been executed during the collection period. This information can be obtained using the `calledServices` operation which returns a list of called services and their valid actions (summarized actions that have been called) in the format:

<servicename> [<valid action>]

Where

<servicename>	Name of Service
<valid actions>	List of valid actions recorded for the service. The table below lists the valid values

Valid Action	Comment
ADD	Service is attempting adding a new instance of an object to the system. For example, adding a to do record.
CHANGE	Service is attempting changes to an existing object in the system.
DEFAULT_ITEM	Service is resetting its values to defaults. For example, by pressing the *Clear* button on the product UI toolbar
DELETE	Service is attempting to delete an existing object
EXECUTE_BO	Service is a business object
EXECUTE_BS	Service is a business service
EXECUTE_LIST	Service is a list based service
EXECUTE_SEARCH	Service is a search
EXECUTE_SS	Service is a service script (including BPA scripts)
READ	Service is attempting to retrieve an object from the system
READ_SYSTEM	Service is a common Oracle Utilities Application Framework based service.
VALIDATE	Service is issuing a validation action

Resetting Statistics

The performance statistics collected represent values since the application was started or when it has been reset. Collection of statistics, without reset, can adversely influence the effectiveness of the statistics over time. It is therefore recommended to reset the statistics on a regular basis (after they are collected for example).

This can be achieved using the `reset` operation from the `PerformanceStatistics` Mbean to effectively zero or blank out the collection statistics.

For example, if the statistics are to be collected on an hourly basis then the reset should occur after the data collection happens per hour.

Note: Any statistics collected during the actual reset operation will not be reflected in the statistics. This situation should have minimal impact on overall statistics.

Database Connection Monitoring

By default, the product uses a common database userid for accessing the information from the connection pools used by the product (via Universal Connection Pool (UCP)). While this

sufficient for execution of the product, it can complicate monitoring individual connections and troubleshooting database issues with individual users or transactions.

It is now possible to show additional details that are inherited from the from the online and Web Services components. The following information is available from the connection and accessible from **v$session**.

Parameter	Online	Web Service
CLIENT_IDENTIFIER[1]	Userid	Userid
MODULE	Service Name	Web Service Name
ACTION	Transaction Type	Transaction Type
CLIENT_INFO	Contents of Database Tag characteristic on User	"Web Service"

For example, the following database query will return the session ids and the users using then at any time:

```
SELECT sid, client_identifier, module, client_info, action FROM V$SESSION;
```

The new information can be used to track sessions using the **v$session** view, use more advanced features of the database and use other database options.

[1] Due to the length limitation on **CLIENT_IDENTIFIER** the value will be the authorization identifier not the authentication identifier.

Configuration

Global Configuration Files

There are a number of configuration files that are global across an environment and also restricted to an environment.

cistab - Global Configuration Files

The **cistab** file is a key configuration file for both the Web application server and the database application server. It is built during the installation process and is used by the product administration utilities to ensure that any output or log files generated by the product are stored in the correct location. It holds the mount points (e.g. directories) used during the installation of the product to hold the product and its log files.

Location of **cistab** file:

Linux/Unix:

`/etc/cistab`

Windows:

`c:\spl\etc\cistab`

A sample cistab file is outlined below:

```
DEV::/spl/DEV:/spl/sploutput/DEV::N
DEMO::/spl/DEMO:/spl/sploutput/DEMO::N
TEST::/spl/TEST:/spl/sploutput/TEST::N
TEST2::d:\spl\TEST2:e:\sploutput\TEST2::N
```

The format of the file is described below:

Position	Usage
1	Environment Name – specified at installation time. It is in UPPER case.
2	Reserved for future use.
3	Directory for the product software and configuration files (the **SPLEBASE** environment variable definition).
4	Directory for the product output files (the **SPLOUTPUT** environment variable definition).
5	Reserved for future use.
6	This flag may be used in custom start up scripts to indicate whether to start the environment at system boot time. Valid values are Y or N. This is the only setting that should be altered after installation.

*Warning! Do not alter the **cistab** file unless instructed to do so by Oracle support personnel unless otherwise directed.*

*Note: For Windows environments it is possible to move the file to alternative drive by setting **%SYSTEMDRIVE%** to an alternative drive prior to running any utilities. For example **set SYSTEMDRIVER=D:** places the **cistab** in **d:\spl\etc**.*

ENVIRON.INI - Environment Configuration File

The ENVIRON.INI file is used by the Web application server and the Business Application Server to define the environment and provide the basis for starting and stopping the environment. The file is created during the installation process and is used to generate other files. This file is maintained using the configureEnv utility provided in the installation.

*Warning! Do not alter the **ENVIRON.INI** manually. Always use configureEnv utility because additional configuration files depend on the settings in this file. If the configurations mismatch, improper operation of the product may occur.*

Location of **ENVIRON.INI** file:

Linux/Unix:

```
$SPLEBASE/etc/ENVIRON.INI
```

Windows:

```
%SPLEBASE%\etc\ENVIRON.INI
```

The file contents are in text format and are of the form:

```
<parameter>=<value>
```

Where:

<parameter>	Name of configuration parameter
<value>	Value of the configuration parameter

For example:

```
…
appViewer=appViewer
DBCONNECTION=jdbc:oracle:thin:@myserver:1521:train
DBDRIVER=oracle.jdbc.driver.OracleDriver
DBNAME=TRAIN
…
```

The settings contained in the **ENVIRON.INI** file are outlined in the table below:

*Note: If **WEB_HELP_FORM_LOGIN_ERROR_PAGE** and/or **WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE** are not specified then they default to the value specified in **WEB_FORM_LOGIN_ERROR_PAGE**.*

*Note: If **WEB_HELP_FORM_LOGIN_PAGE** and/or **WEB_APPVIEWER_FORM_LOGIN_PAGE** are not specified then they default to the value specified in **WEB_FORM_LOGIN_PAGE**.*

*Note: If **WEB_APPVIEWER_ROLE_NAME** and/or **WEB_APPVIEWER_PRINCIPAL_NAME** are not specified they are default to **WEB_ROLE_NAME** and **WEB_PRINCIPAL_NAME** respectively.*

Extracting Information from ENVIRON.INI for Scripts

It is possible to write your own calls to the **ENVIRON.INI** using the same utilities used by the product to get values of configuration parameters for your own utilities. Do not hardcode values that can be obtained from **ENVIRON.INI**.

To obtain values of parameters use the command line:

Linux/Unix:

```
perl $SPLEBASE/bin/getconfvalue.plx -k <parameter>
```

Windows:

```
perl %SPLEBASE%\bin\getconfvalue.plx -k <parameter>
```

Where:

<parameter>	Name of configuration parameter from ENVIRON.INI you desire to get the value of.

For example:

ENVIRON.INI content:

```
…
DBNAME=TRAIN
…
```

Example call:

```
$ export DB=`perl $SPLEBASE/bin/getconfvalue.plx -k DBNAME`
$ echo $DB
TRAIN
```

Note: If the value is NOT set or the key is invalid the value of the call is null or blank.

Server Jar File (ouaf_jar_versions.txt)

Note: This configuration file is used for internal purposes and should not be altered unless instructed to do so by Oracle Support.

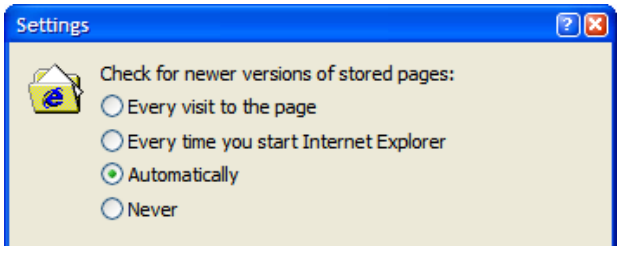
Note: Additional external jar files used for customizations do not need to be added to this file. This is used for base product verification only.

The java component of the product uses a number of industry standard jar files that are provided or used by the product. The **etc/ouaf_jar_versions.txt** lists the jar file that is used and the required version used by the version of the product installed. This file is used at installation and runtime for integrity checks. If you wish to determine what version of an external jar is used then refer to this information file.

Web Browser Configuration

The product is browser based (browsers, versions and platforms are documented in the Installation Guide for your platform. Additionally the following settings are applicable to the browser:

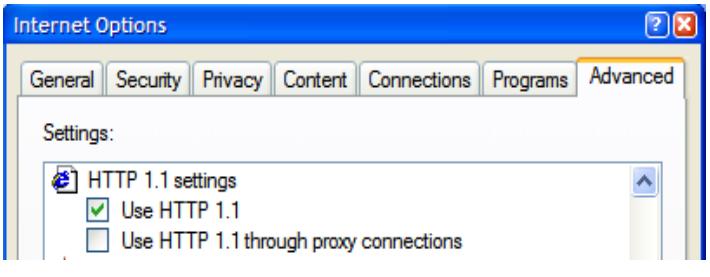
- **Microsoft Internet Explorer** - Cache settings need to be *Every visit to the page* or *Automatically*. For non-production it is recommended to be set to *Every visit to the page* or *Automatically*. For production it is recommended to be set to *Automatically* to fully exploit performance caching.



- **Mozilla Firefox** – Use the default settings with the browser for the browser.

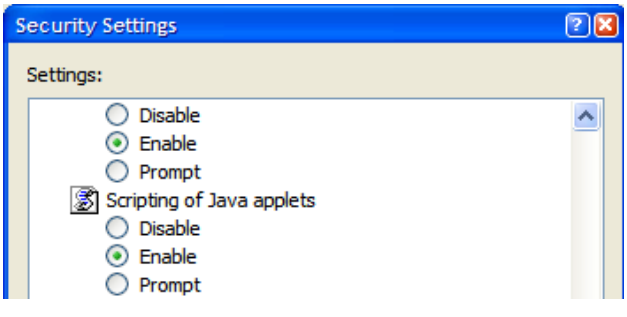
Note: Clearing the cache upon exit will clear the cached screens of the product as well

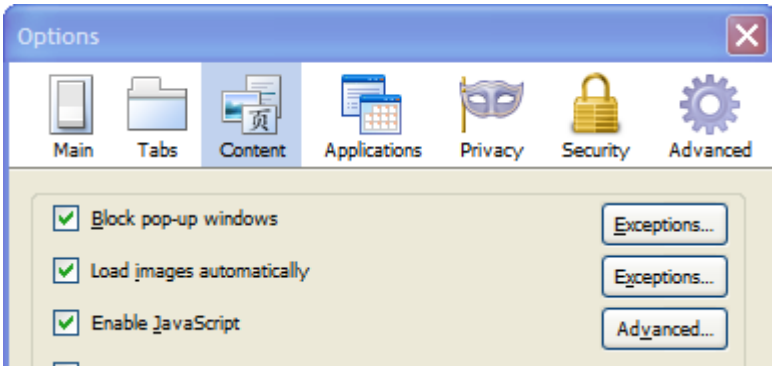
- The product requires support for the HTTP 1.1 protocol to support compression and client cache management.



Note: If a network proxy is used then "Use HTTP 1.1 through proxy connections" may need to be selected as well.

- The product uses Java scripting for user interactivity therefore *Scripting of Java Applets* (IE) and "Enable Java Script" (Firefox) must be enabled.





- The product uses popup windows for searches, therefore popup blockers should be configured to allow popups from the product Web application server hosts.
- Set your browser cache size to a reasonable size to hold the cached pages as needed.

Web Application Server Configuration

Caveat

The product supports a number of J2EE Web application servers. Each J2EE Web application server is configured differently and has additional options (clustering, logging etc) that can be used. This document is written neutral to the differences of each J2EE Web application server. Refer to the documentation provided with the J2EE Web application servers for the location of specific configuration settings discussed in this section as well as advanced settings supported.

Web Application Server Concepts

Each Web application server has a number of levels and each uses different terminology. The following "neutral" terminology will be used:

- The software exists on a physical machine.
- An installation of the Web Application Software is called an instance. Typically one instance of the software exists on a machine but you can have more than one installed.
- Within an instance you can define a server. This is also called a Java "container" which will house one or more J2EE applications. You will have at least one server per environment. A server uses one Java Virtual Machine (JVM).
- Within a server is the J2EE application. It can be a single J2EE application or multiples depending on the Web application server supported.

The Web application server you use may have different terminology for these same concepts. For the remainder of this section we will use the above terminology.

Web Applications

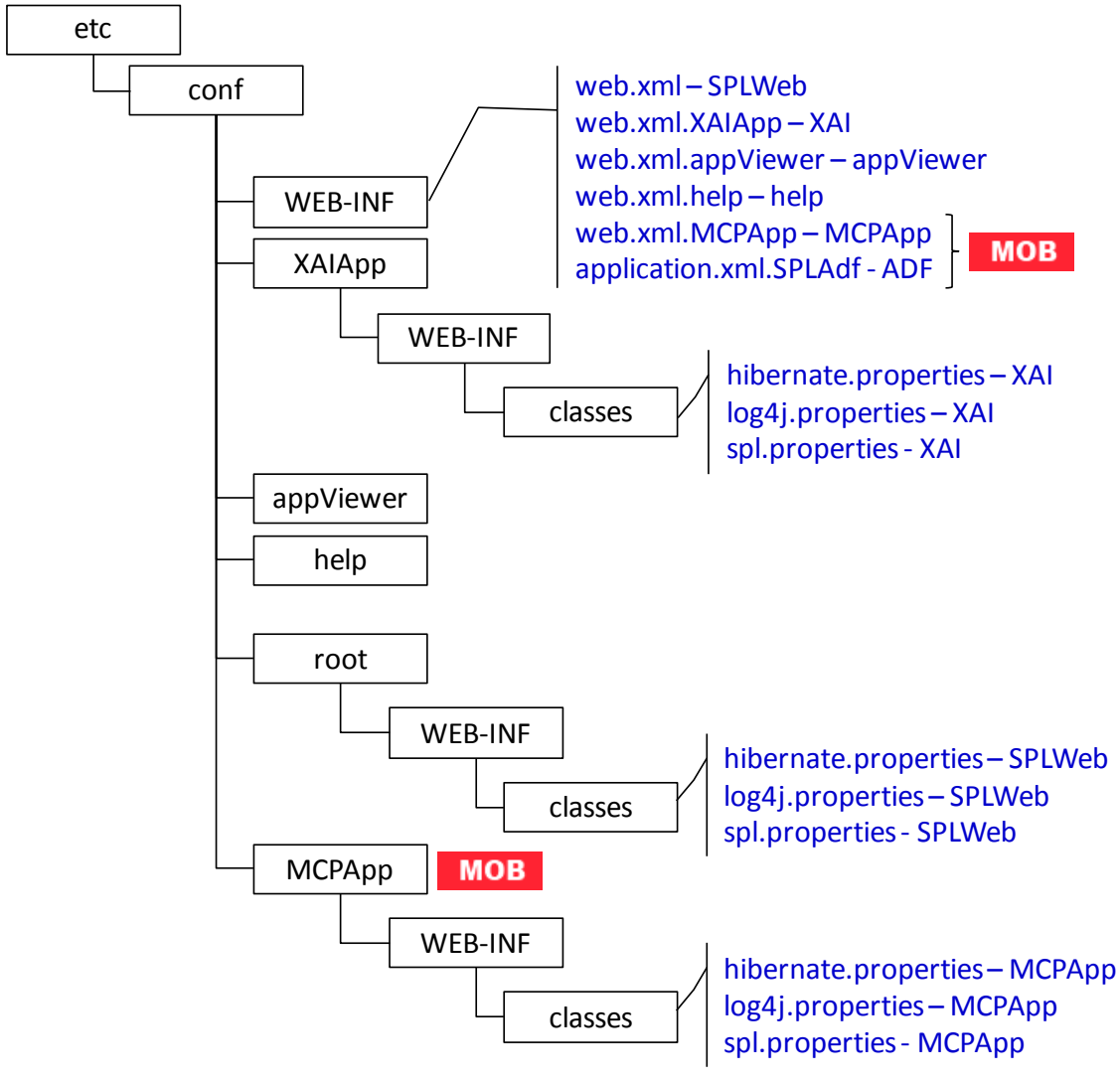
The product is deployed as a set of Web applications within the Web applications server:

- **root** – This is the product itself is installed.
- **XAIApp** – This is the Web services adapter component.
- **appViewer** – An Application Viewer which contains a data dictionary and source viewer.
- **help** – Online Help.
- **MCPApp** – Mobile Connection Platform (MOB only)

Each of these J2EE Web Applications has its own configuration files and are combined together when the product is "built" into a WAR/EAR file by the initialSetup utility.

Web Application Server Configuration Files

Within each J2EE Web Application within the J2EE Web application server has it's own configuration files. These files are typically "embedded" within the WAR/EAR files deployed with the product following the J2EE specification. In terms of configuration, the product structure within the WAR/EAR file looks like the following:



Location	Contents	Configuration Files
WEB-INF	J2EE Application Descriptor for each application	web.xml – J2EE Application Descriptor
root/WEB-INF/classes	Application Configuration files for online application	log4j.properties – Logging Configuration spl.properties – Product configuration settings
XAIApp/WEB-INF/classes	Application Configuration files for Web Services Adapter	log4j.properties – Logging Configuration spl.properties – Product configuration settings
MCPApp/WEB-INF/classes	MCP Configuration MOB	log4j.properties – Logging Configuration spl.properties – Product

Location	Contents	Configuration Files
		configuration settings©

web.xml – J2EE Application Descriptor

The Web deployment descriptor editor lets you specify deployment information for modules created in the Web development environment. The information appears in the **web.xml** file. The **web.xml** file for a Web project provides information necessary for deploying a Web application module. It is used in building a WAR/EAR file from a project.

The Web Application is controlled by a configuration file that holds behavioral information for the applications. Refer to http://jcp.org/en/jsr/detail?id=109 for more details of the format. For example:

```
...
<env-entry>
      <description>Value of HTTP 1.1 max-age header parameter for
JSPs</description>
      <env-entry-name>maxAge</env-entry-name>
      <env-entry-value>28800</env-entry-value>
      <env-entry-type>java.lang.Integer</env-entry-type>
   </env-entry>
   <env-entry>
      <description>How long to cache drop down values in
seconds</description>
      <env-entry-name>fieldValuesAge</env-entry-name>
      <env-entry-value>3600</env-entry-value>
      <env-entry-type>java.lang.Integer</env-entry-type>
   </env-entry>
   <env-entry>
      <description>Is this a development environment</description>
      <env-entry-name>isDevelopment</env-entry-name>
      <env-entry-value>false</env-entry-value>
      <env-entry-type>java.lang.Boolean</env-entry-type>
   </env-entry>
   <env-entry>
      <description>Preload ALL Pages</description>
      <env-entry-name>preloadAllPages</env-entry-name>
      <env-entry-value>false</env-entry-value>
      <env-entry-type>java.lang.Boolean</env-entry-type>
   </env-entry>
   <env-entry>
      <description>Disable preloading of Pages</description>
      <env-entry-name>disablePreload</env-entry-name>
      <env-entry-value>false</env-entry-value>
```

```
        <env-entry-type>java.lang.Boolean</env-entry-type>
    </env-entry>
...
```

For application specific entries refer to the web.xml parameter descriptions.

Note: It is highly recommended that you do not change this configuration file by extracting the configuration file from the WAR/EAR file using Java utilities, making the change manually and rebuilding the WAR/EAR file. Use initialSetup – Maintain Configuration Settings to build the WAR/EAR file as documented in Web application server Configuration Process

log4j.properties – Logging Configuration

Note: This log file should not be altered unless specified. The generated configuration file has all the recommended settings for all sites.

The product uses the *log4j* Java classes to centralize all log formats into a standard format. The details of the configuration settings and *log4j* itself are available at http://logging.apache.org/log4j/ or http://en.wikipedia.org/wiki/Log4j.

spl.properties – Product configuration settings

The product Web Application has a specific number of settings outside of the J2EE specification to control the internals of the product. This file exists as similar files exist for all modes of operation of the product (for example, Batch can be run outside the J2EE Web application server). Because of this a common configuration standard was adopted:

For a description of all settings in the `spl.properties` file refer to the `spl.properties` parameter descriptions.

weblogic.xml – WebLogic Extensions

Note: This configuration file only applies to Oracle WebLogic implementations.

For backward compatibility with Oracle WebLogic environments, an additional Oracle WebLogic configuration file **weblogic.xml** is generated and used to influence the Oracle WebLogic Server to exhibit additional behavior (targeted for development primarily).

Parameter	Context	Source
`context-root`	The context-root element defines the context root of this stand-alone Web application. If the Web application is part of an EAR, not stand-alone, specify the context root in the EAR's **web.xml** file. A context-root setting in **web.xml** takes precedence over context-root setting in **weblogic.xml**.	Defaults from template

Parameter	Context	Source
java-charset-name	Specifies the Java character set to use.	Defaults from template (UTF-8)
page-check-seconds	Determines the interval at which a server checks to see if JSP files in a Web application have changed and need recompiling. Used for development	Derived from WEB_WLPAGECHECKSECONDS parameter from ENVIRON.INI
prefer-web-inf-classes	Loading of web classes from the WEB-INF are loaded in preference to system or Oracle WebLogic classes. Defaulted to false.	Defaults from template
resource-path	A path which, if included in the URL of a request, signals Oracle WebLogic Server to use the Java character set specified by java-charset-name.	Defaults from template
servlet-reload-check-secs	Defines whether an Oracle WebLogic Server will check to see if a servlet has been modified, and if it has been modified, reloads it. The -1 value tells the server never to check the servlets, 0 tells the server to always check the servlets, and the default is to check each 1 second. A value specified in the console will always take precedence over a manually specified value.	Defaults from template
url-rewriting-enabled	Provides methods for configuring a J2EE web application that is deployed on an Oracle WebLogic Server instance. Oracle WebLogic Server instantiates this interface only when you deploy a web application. This interface can configure web applications that are deployed as a WAR file or an exploded directory.	Defaults from template (false)

Note: This configuration file is not usually altered by an implementation as it applies to development (SDK) platforms only. It is documented for completeness here.

Example:

```
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90">
   <session-descriptor>
      <url-rewriting-enabled>false</url-rewriting-enabled>
   </session-descriptor>
   <jsp-descriptor>
      <page-check-seconds>43200</page-check-seconds>
   </jsp-descriptor>
  <container-descriptor>
    <servlet-reload-check-secs>-1</servlet-reload-check-secs>
    <prefer-web-inf-classes>true</prefer-web-inf-classes>
  </container-descriptor>
  <charset-params>
    <input-charset>
      <resource-path>/*</resource-path>
      <java-charset-name>UTF-8</java-charset-name>
    </input-charset>
  </charset-params>
  <context-root>/</context-root>
</weblogic-web-app>
```

application.xml – ADF Application configuration

Note: This configuration file only applies to Oracle WebLogic and Oracle ADF implementations.

To use the Oracle Application Development Framework (ADF) integration the ADF components need to be deployed to a predefined ADF container. The definition of this container is controlled by the J2EE standard **application.xml** file.

Parameter	Context	Source
context-root	ADF context root used for calls	Set to **WEB_CONTEXT_ROOT/adf**
display-name	Specifies the application display name	Set to **SPLAdf**
web-uri	Defines location of WAR file	Set to **SPLAdf**

Example:

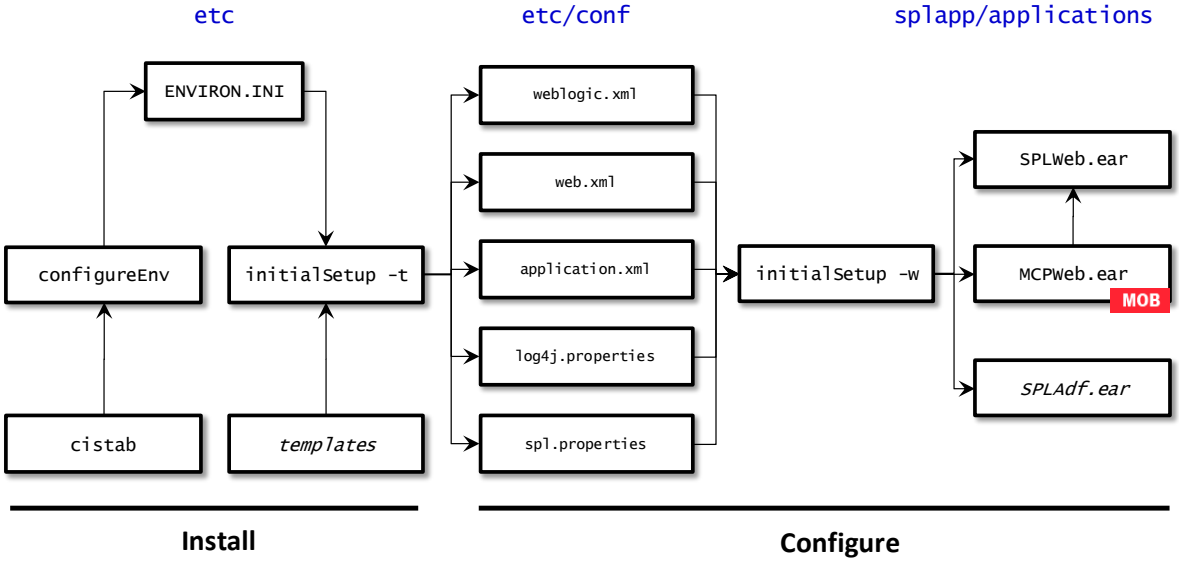
```
<?xml version = '1.0'?>
<application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd" version="5"
xmlns="http://java.sun.com/xml/ns/javaee">
  <display-name>SPLAdf</display-name>
  <module>
    <web>
```

```
      <web-uri>SPLAdf</web-uri>
      <context-root>demo/adf</context-root>
    </web>
  </module>
</application>
```

Note: This file should not be altered unless instructed by Oracle Support.

Web Application Server Configuration Process

To configure the Web application server during the installation process and post-installation then the following process should be used:



- The configureEnv utility is used during installation time and can be used post implementation to set parameters in the ENVIRON.INI. If any parameters are derived or set from the ENVIRON.INI (see "Source" column in the relevant section) then the configureEnv utility should be used to maintain them.

  Note: The configureEnv utility should be used to make any changes to the ENVIRON.INI. Manual changes to this configuration file are not recommended.

- After the ENVIRON.INI has been set or altered, the settings must be reflected in the relevant configuration files used by the Web application server by running the initialSetup utility:
  - web.xml – J2EE Application Descriptor
  - log4j.properties – Logging Configuration
  - spl.properties – Product configuration settings
- The utility uses the templates from the **templates** directory to create substituted copies of these files in a standard location. The table below lists the configuration file, the templates used from the **templates** directory and the final configuration built during the initial configuration process:

Configuration File	Destination
Online Application (root)	

web.xml
Template:
`web.xml.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

spl.properties
Template:
`spl.properties.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/root/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\root\WEB-INF\classes`

log4j.properties
Template:
`log4j.properties.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/root/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\root\WEB-INF\classes`

weblogic.xml
Template:
`weblogic.xml.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/root/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\root\WEB-INF`

Web Services Adapter (XAIApp)	

web.xml (*web.xml.XAIApp*)
Template:
`web.xml.XAIApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

spl.properties
Template:
`spl.properties.XAIApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/XAIApp/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\XAIApp\WEB-INF\classes`

log4j.properties
Template:
`log4j.properties.XAIApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/XAIApp/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\XAIApp\WEB-INF\classes`

Configuration File	Destination

weblogic.xml
Template:
`weblogic.xml.XAIApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/XAIApp/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\XAIApp\WEB-INF`

Application Viewer (appViewer)	

web.xml (*web.xml.appViewer*)
Template:
`web.xml.appViewer.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

Help Application (help)	

web.xml (*web.xml.help*)
Template:
`web.xml.help.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

MCP Application (MCPApp) MOB	

web.xml *(web.xml.MCPApp)*
Template:
`MWM_web.xml.MCPApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

spl.properties
Template:
`MWM_spl.properties.MCPApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/MCPApp/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\MCPApp\WEB-INF\classes`

log4j.properties
Template:
`MWM_log4j.properties.MCPApp.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/MCPApp/WEB-INF/classes`
	Windows:
	`%SPLEBASE%\etc\conf\MCPApp\WEB-INF\classes`

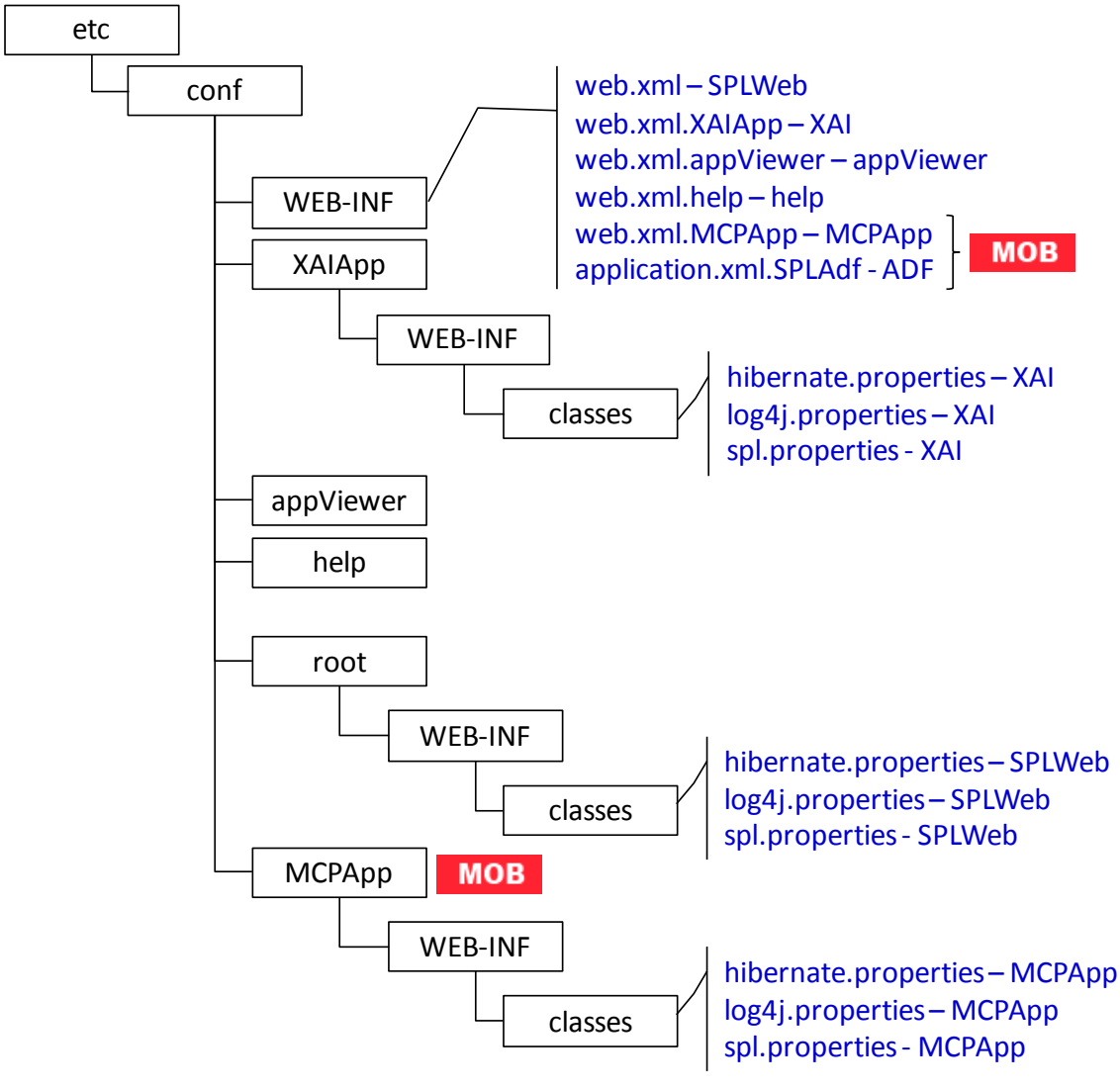
SPLAdf Application (ADF Integration)	

application.xml (*application.xml.SPLAdf*)
Template:
`MWM_application.xml.SPLAdf.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

Configuration File	Destination

weblogic.xml (*weblogic.xml.SPLAdf*)
Template:
`MWM_weblogic.xml.SPLAdf.template`
	Linux/Unix:
	`$SPLEBASE/etc/conf/WEB-INF`
	Windows:
	`%SPLEBASE%\etc\conf\WEB-INF`

The locations of the configuration files can be summarized in the following figure:



- At this point you may perform manual changes to the above files to parameters not implemented in the ENVIRON.INI.

  Note: Any manual changes are overwritten after running the initialSetup utility unless the change is reflected in the appropriate template (see Implementing Custom Templates for more information). Backups should be made of any changes and then manually reapplied to reinstate all manual changes.

- To reflect configuration changes into the product Web Applications the initialSetup utility with the **–w** option must be executed. This will build the necessary WAR/EAR files to be deployed into the J2EE Web application server. This step is optional if configuration overrides are in use.

Depending on the architecture, the initialSetup will generate one or more EAR files. Refer to Business Application Server Configuration for a description of the EAR files.

At this point the product Web Applications are ready for deployment into the J2EE Web application server.

Quick Reference Guide for Web Application Server Configuration

To make configuration changes to the Web Application Server component of the product uses the following Quick Reference Guide to identify which process should be used:

- If the change is to any setting contained in the ENVIRON.INI for the Web Application Server then you must run the following utilities in the order indicated:
  1. Execute the configureEnv utility to reflect the parameter change in the ENVIRON.INI.
  2. Execute the initialSetup utility (with the **–t** option) to rebuild the configuration files using the ENVIRON.INI and provided template files. This will reset the configuration to the contents of the base template files or custom template (if used).
  3. Any configuration changes that are overridden by templates (base or custom) must be manually reapplied (if necessary).
  4. Execute the initialSetup utility (with the **–w** option) to implement the configuration files in the product Web Application Server files. This step is not necessary of you are using configuration overrides.
- If the change is to any setting not contained in the ENVIRON.INI for the Web application server but is in the configuration files for the Web Application Server then you must run the following utilities in the order indicated:
  1. Make any manual changes to the relevant configuration files.
  2. Execute the initialSetup (with the **-w** option) utility to implement the configuration files in the product Web Application Server files. This step is not necessary of you are using configuration overrides.

User Interface Backward Compatibility Settings

The following settings are used for backward compatibility of the User interface for customers upgrading from an Oracle Utilities Application Framework V2.1 based product (values of true emulate V2.1 user interface behavior for UI Maps):

- **`spl.runtime.compatibility.uiMapDisableInputValue`** – By default, if the XML schema has an input value with a default then setting this value to false will cause the product to set this value to blank and ignore the default value for add mode in all UI Maps. Setting of this value to true will cause the default to be displayed in the input field for add mode in all UI Maps. The default value for this parameter is false.
- **`spl.runtime.compatibility.uiMapDropdownSelectFirstValue`** – By

default, dropdown widgets on UI Maps are defaulted to no value to force the user to select a value. By setting this parameter to true, forces all dropdowns on all UI Maps to automatically default to the first value in the dropdown list. By setting this value to false, the default, the UI Maps will have blank values as the default value for the dropdowns.

- `spl.runtime.compatibility.uiMapDisableTitle` - By default UI Maps contain a rendered title section. By setting this parameter to true, the title sections for all UI Maps are not automatically rendered. By setting this parameter to false, the default, title sections are rendered automatically for all UI Maps.
- `spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs` – By default screen elements have unique ids for reference, including individual records in lists or queries. By setting this value to false, the default, the framework will generate unique ids for ADA compliance. If customizations from past releases have issues with these unique ids then setting the value to true will revert to behavior available in past releases of the product.

Note: It is recommended to leave the default value, false, for these parameters unless otherwise required or instructed by Oracle Support.

Web Application Server Deployment Process

After the configuration of the Web Application is complete (as outlined in Web application server Configuration Process) the final step to implement the product technically is to deploy the product within the J2EE Web application server.

There are three methods of deploying the product within the J2EE Web application server:

1. Use the deployment utilities provided on the console of the J2EE Web application server. The WAR/EAR files that are available under `$SPLEBASE/splapp/applications` (or `%SPLEBASE%\splapp\applications` for Windows) can be manually deployed using the console. Refer to the Installation Guide for specific platform instructions and the administration guide for the J2EE Web application server.

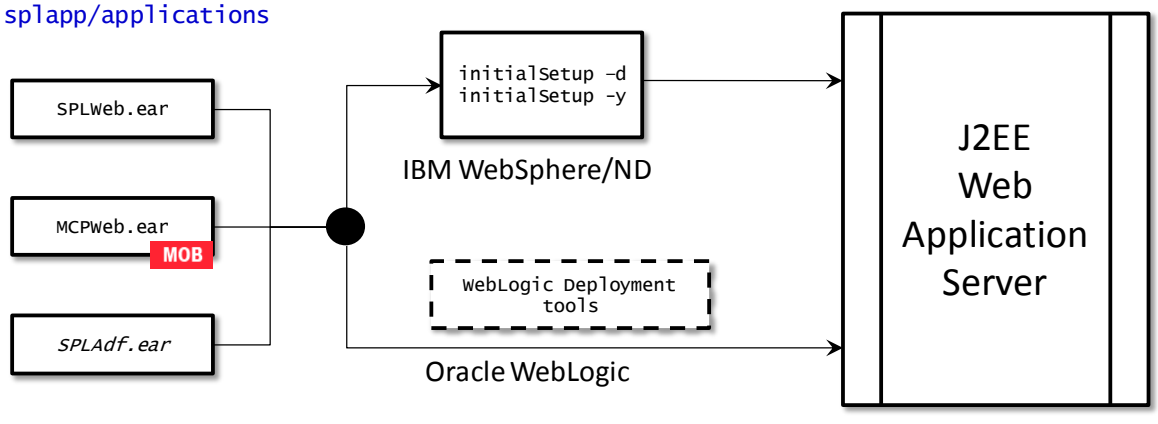
   Note: This is the only method that can be used if virtual Web application servers are used with the product.

2. Use the deployment utilities provided on the command line of the J2EE Web application server. The WAR/EAR files that are available under `$SPLEBASE/splapp/applications` (or `%SPLEBASE%\splapp\applications` for Windows) can be manually deployed using the command line utilities supplied with your J2EE Web application server. Refer to the Installation Guide for specific platform instructions and the administration guide for the J2EE Web application server.

3. A number of specific utilities for J2EE Web applications are provided with the product to deploy the Web Application to the J2EE Web application server. These call the same utilities provided in Option 2 but are provided with the product.

Note: This section will outline Option 3 only.

A number of utilities are provided in the *bin* directory of the product to deploy the product to the J2EE Web application server. These utilities are outlined below:



Deploy

- For the IBM WebSphere or IBM WebSphere ND platform, use the initialSetup utility (with the **–d** or **-y** options) utility. This will call the relevant IBM WebSphere utility to perform the deployment.

  *Note: The **–y** option allows for a decoupled installation on IBM WebSphere. On Oracle WebLogic the console may be used to configure individual elements to achieve the same functionality.*

- For Oracle WebLogic, there are two options:
  - *Native Mode* – Use the WebLogic console or WLST to deploy/redeploy the EAR files.
  - *Embedded Mode* - No additional deployment is necessary as the product automatically detects Oracle WebLogic and allows Oracle WebLogic to read the WAR/EAR files directly.

These utilities will attempt to deploy the Web Applications within the J2EE Web application server as follows:

J2EE Web application server	Deployment details
Oracle WebLogic	Deployed to **WEB_CONTEXT_ROOT** application by default using **WEB_WLSYSUSER** and **WEB_WLSYSPASS** from the ENVIRON.INI as administration credentials.
IBM WebSphere	Deployed to **WEB_APP** Application on **WEB_SVRNAME** server by default using **WEB_WASUSER** and **WEB_WASPASS** from ENVIRON.INI as administration credentials.
IBM WebSphere ND	Deployed to **WEB_APP** Application on **WEB_SVRNAME** server on **WEB_NODENAME** by default using **WEB_WASUSER** and **WEB_WASPASS** from ENVIRON.INI as administration credentials.

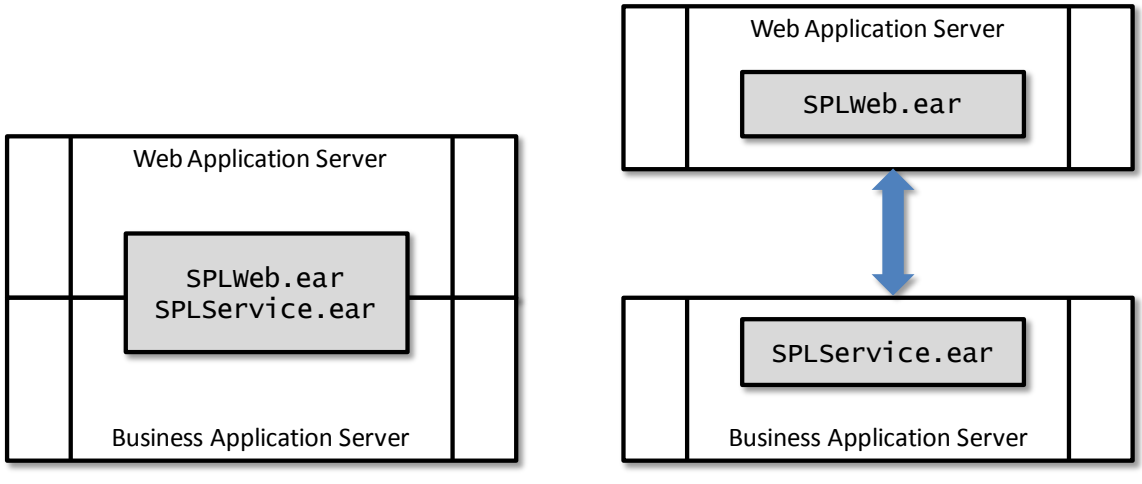
The Web Application should be available from the Web Application Server.

Business Application Server Configuration

It is possible for the Business Application Server logic to be separated from the Web Application Server component. Essentially the product has been split into TWO distinct EAR files:

- **SPLWeb.ear** – This contains the Web application server component for the product.
- **SPLService.ear** - This contains the Business Application Server component for the product.

There are two modes of installation:



Local install Decoupled install

- "**Local**" **Installation** (also applicable to *expanded* installations for Development environments) - The Web application server and Business Application are on the same instance of the J2EE Web application server. This is the *default* behavior of the product for backward compatibility. If this is the mode installed then for configuration the process is a combination of the Web Application Server and Business Application Server configuration and deployment process.

  Note: Local installations are only supported on development platforms and Oracle WebLogic installations only.

- **Decoupled Installation** – The Business Application Server is on a separate instance of the J2EE Web application server. This may be the same machine or different machines. In this case the Web Application Server and Business Application Server are managed and configured separately. To perform a decoupled installation the following must be performed:

  1. The product is installed on the machines housing the Web Application Server and Business Application Server.
  2. A set of "servers" within one or more instances of the J2EE Web Application Server must be created to house the Web Application Server and Business

Application Server separately. This can be on the same machine or across machines.

3. The Web Application Server and Business Application Server are configured as outlined in Web Application Server Configuration and Business Application Server Configuration.

4. The WAR/EAR files generated are deployed separately with the **SPLWeb.ear** EAR file deployed to the Web application server as outlined in Web Application Server Deployment Process and **SPLService.ear** EAR file deployed to the Business Application Server as outlined in Business Application Server Deployment Process.

Note: For customers using Oracle ExaLogic, Oracle highly recommend that local installations be used for performance reasons.

Business Application Server Concepts

As mentioned previous the Business Application Server component can be deployed within a separate instance of the J2EE Web Application server Software. This effectively allows the Business Application Server to be on separate hardware for architectures where this is a requirement. Typically this separation is implemented for a number of reasons:

- The site has an architectural principle for separating the Business Application Server and Web application server.
- The site prefers to optimize the individual servers for the individual tiers rather than having to compromise when two or more tiers are on the same platform.

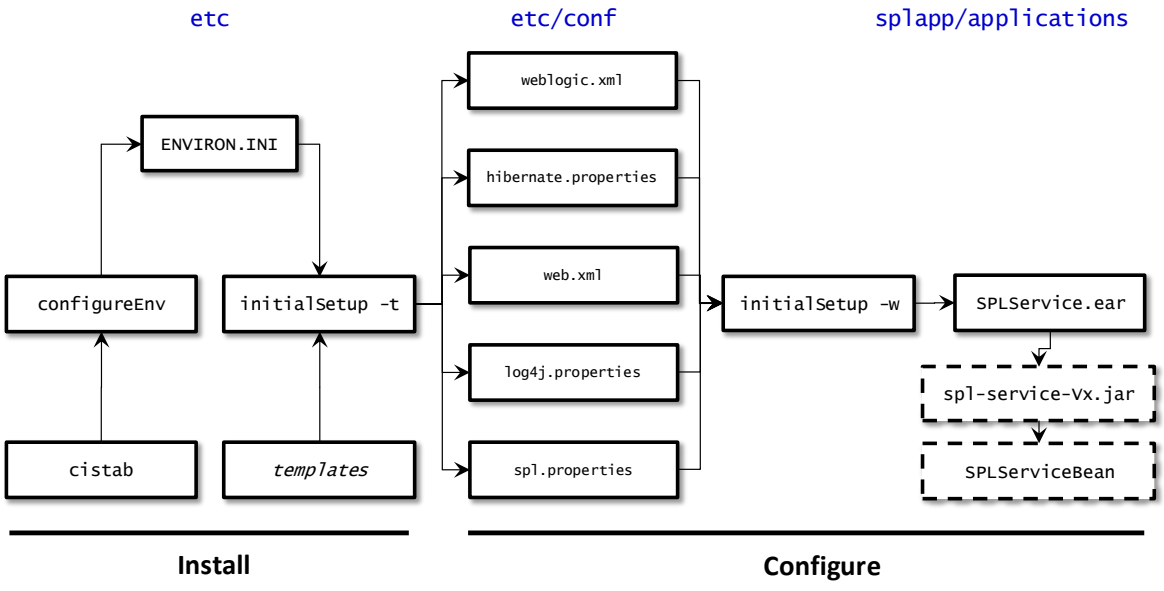
The Business Application Server was designed to fit within the same concepts as the Web Application Server. The main differences are:

- Enterprise Java Beans (stateless) are used in the Business Application Server instead of Java Server Pages as used in the Web application server. The name of the EJB is `spl-servicebean-<version>.jar` (where *`<version>`* is the version of the product e.g. 2.0.0).
- Database connectivity is configured in the Business Application Server.

The rest of this section will outline the differences specifically for the Business Application Server.

Business Application Server Configuration Process

To configure the Business Application Server during the installation process and post-installation then the following process should be used:



- The configureEnv utility is used during installation time and can be used post implementation to set parameters in the ENVIRON.INI. If any parameters are derived or set from the ENVIRON.INI (see "Source" column in the relevant section) then the configureEnv utility should be used to maintain them.

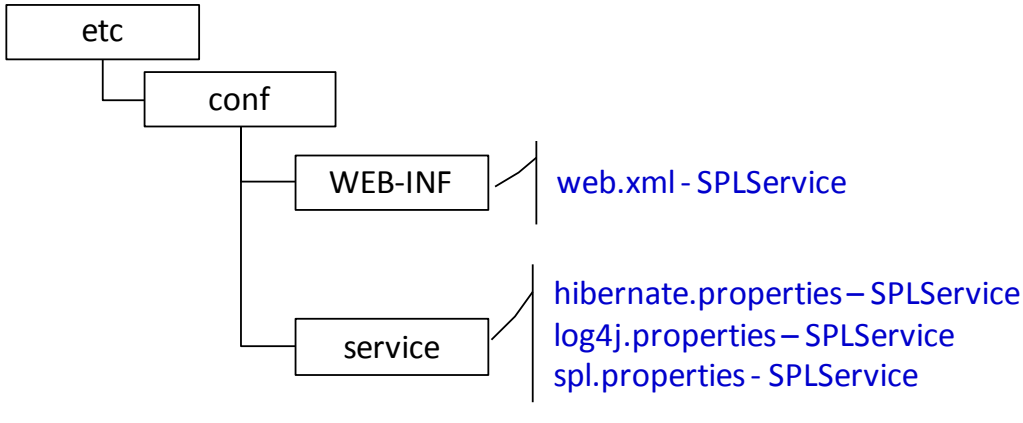
  Note: The configureEnv utility should be used to make ANY changes to the ENVIRON.INI. Manual changes to this configuration file are not recommended.

- After the ENVIRON.INI has been set or altered, the settings must be reflected in the relevant configuration files used by the Business Application Server by running the initialSetup:
  - log4j.properties – Logging Configuration
  - spl.properties – Product configuration settings
  - hibernate.properties – Database connectivity properties
  - web.xml – J2EE Application Descriptor
- The utility uses the templates from the **templates** directory to create substituted copies of these files in a standard location:

Configuration File	Destination
Service Bean	
web.xml Template: web.xml.template	Linux/Unix: $SPLEBASE/etc/conf/WEB-INF Windows: %SPLEBASE%\etc\conf\WEB-INF
spl.properties Template: spl.properties.service.template	Linux/Unix: $SPLEBASE/etc/conf/service $SPLEBASE/splapp/businessapp/properties

Configuration File	Destination
	Windows: %SPLEBASE%\etc\conf\service %SPLEBASE%\splapp\businessapp\properties
log4j.properties Template: log4j.properties.service.template	Linux/Unix: $SPLEBASE/etc/conf/service $SPLEBASE/splapp/businessapp/properties Windows: %SPLEBASE%\etc\conf\service %SPLEBASE%\splapp\businessapp\properties
hibernate.properties Template: hibernate.properties.web.template	Linux/Unix: $SPLEBASE/etc/conf/root/WEB-INF/classes $SPLEBASE/etc/conf/XAIApp/WEB-INF/classes $SPLEBASE/etc/conf/service Windows: %SPLEBASE%\etc\conf\root\WEB-INF\classes %SPLEBASE%\etc\conf\XAIApp\WEB-INF/classes %SPLEBASE%\etc\conf\service

The locations of the configuration files can be summarized in the following figure:



- At this point you may perform manual changes to the above files to parameters not implemented in the ENVIRON.INI.

Note: Any manual changes are overwritten after running the initialSetup utility unless the change is reflected in the appropriate template (see custom templates for more information). Backups should be made of any changes and then manually reapplied to reinstate all manual changes.

- To reflect configuration changes into the product Business EJB Applications the

initialSetup utility, with the **–w** option, must be executed. This will build the necessary **spl-servicebean-<*version*>.jar** (where **<*version*>** is the version of the product used) and the **SPLService.ear** EAR file to be deployed into the J2EE Web application server. This step is optional if configuration overrides are in use (refer the discussion of allowing the externalization of configuration settings for alternative methods).

Depending on the architecture used, the initialSetup will generate one or more EAR files.

At this point the product Business Applications are ready for deployment into the J2EE Web application server.

Quick Reference Guide for Business Application Server Configuration

To make configuration changes to the Business Application Server component of the product uses the following Quick Reference Guide to identify which process should be used:

- If the change is to any setting contained in the ENVIRON.INI for the Business Application Server then you must run the following utilities in the order indicated:
  1. Execute the configureEnv utility to reflect the parameter change in the ENVIRON.INI.
  2. Execute the initialSetup utility (with the **–t** option) to rebuild the configuration files using the ENVIRON.INI and provided template files. This will reset the configuration to the contents of the base template files or custom template (if used.
  3. Any configuration changes that are overridden by templates (base or custom) must be manually reapplied (if necessary).
  4. Execute the initialSetup utility (with the **–w** option) to implement the configuration files in the product Business Application files. This step is not necessary of you are using configuration overrides
- If the change is to any setting not contained in the ENVIRON.INI for the Business Application Server but is in the configuration files for the Business Application Server then you must run the following utilities in the order indicated:
  1. Make any manual changes to the relevant configuration files.
  2. Execute the initialSetup, with the **-w** option, utility to implement the configuration files in the product Business Application Server files. This step is not necessary of you are using configuration overrides.

Business Application Server Deployment Process

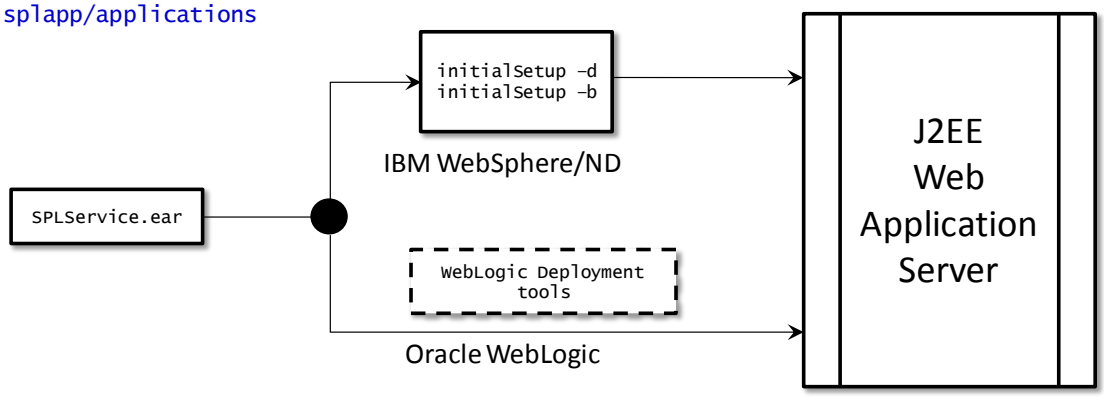
After the configuration of the Business Application Server is complete (as outlined in Business Application Server Configuration Process) the final step to implement the product technically is to deploy the product within the J2EE Web application server.

There are three methods of deploying the product within the J2EE Web application server:

- Use the deployment utilities provided on the console of the J2EE Web application server. The WAR/EAR files that are available under **$SPLEBASE/splapp/applications** (or **%SPLEBASE%\splapp\applications** for Windows) can be manually deployed using the console. Refer to the Installation Guide for specific platform instructions and the administration guide for the J2EE Web application server.
- Use the deployment utilities provided on the command line of the J2EE Web application server. The WAR/EAR files that are available under **$SPLEBASE/splapp/applications** (or **%SPLEBASE%\splapp\applications** for Windows) can be manually deployed using the J2EE Web application server vendor supplied deployment command line utilities. Refer to the Installation Guide for specific platform instructions and the administration guide for the J2EE Web application server.
- A number of specific utilities for J2EE Web Application are provided with the product to deploy the EJB Application to the J2EE Web application server. These call the same utilities provided in the previous option but are provided with the product.

This section will outline the latter option.

A number of utilities are provided in the *bin* directory to deploy the product to the J2EE Web application server. These utilities are outlined below:



- For the IBM WebSphere/WebSphere ND platform, use the initialSetup utility (with the **–d** or **–b** options). This will call the relevant IBM provided utility to deploy the WAR/EAR files into the IBM WebSphere instance.

  *Note: The **–b** option allows for a decoupled installation on IBM WebSphere. On Oracle WebLogic the console may be used to configure individual elements to achieve the same functionality.*

- For Oracle WebLogic, there are two options:
  - *Native Mode* – Use the WebLogic console or WLST to deploy/redeploy the EAR files.
  - *Embedded Mode* - No additional deployment is necessary as the product automatically detects Oracle WebLogic and allows Oracle WebLogic to read the

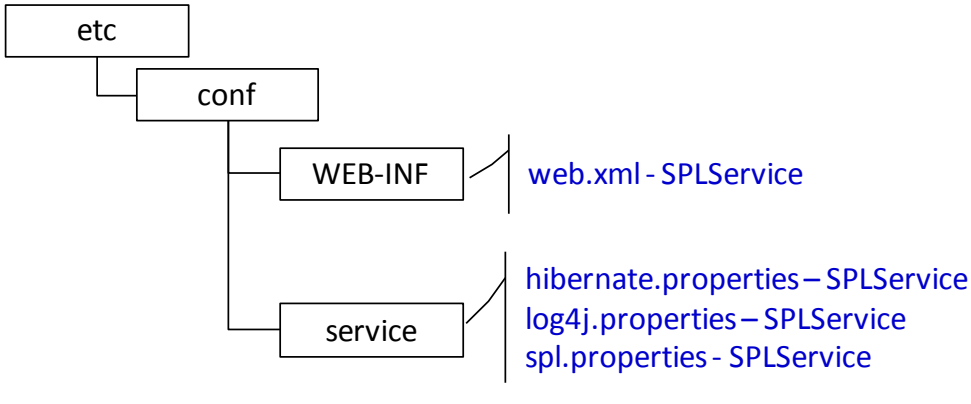
WAR/EAR files directly.

These utilities will attempt to deploy the EJB Applications within the J2EE Web application server as follows:

J2EE Web application server	Deployment details
Oracle WebLogic	Deployed to **WEB_CONTEXT_ROOT** application by default using **WEB_WLSYSUSER** and **WEB_WLSYSPASS** from the ENVIRON.INI as administration credentials.
IBM WebSphere	Deployed to **BSN_APP** Application on **BSN_SVRNAME** server by default using **WEB_WASUSER** and **WEB_WASPASS** from ENVIRON.INI as administration credentials.
IBM WebSphere ND	Deployed to **BSN_APP** Application on **BSN_SVRNAME** server on **BSN_NODENAME** by default using **WEB_WASUSER** and **WEB_WASPASS** from ENVIRON.INI as administration credentials.

Business Application Server Configuration Files

Each J2EE Web Application within the J2EE Web application server has its own configuration files. These files are typically *embedded* within the WAR/EAR files deployed with the product following the J2EE specification (refer the discussion of allowing the externalization of configuration settings for alternative methods). In terms of configuration, the product structure within the WAR/EAR file looks like the following:



Location	Contents	Configuration Files
WEB-INF	J2EE Application Descriptor for Business Application Server	web.xml – J2EE Application Descriptor
service	Application Configuration files for Business Application Server	log4j.properties – Logging Configuration hibernate.properties –

Location	Contents	Configuration Files
		Database connectivity properties spl.properties – Product configuration settings

web.xml – J2EE Application Descriptor

The Web deployment descriptor editor lets you specify deployment information for modules created in the Web development environment. The information appears in the *web.xml* file. The *web.xml* file for a Web project provides information necessary for deploying a Web application module. It is used in building a WAR/EAR file from a project.

The Business Application is controlled by a configuration file that holds behavioral information for the applications. Refer to http://jcp.org/en/jsr/detail?id=109 for more details of the format. For example:

```
…
   <env-entry>
      <description>How long to cache drop down values in
seconds</description>
      <env-entry-name>fieldValuesAge</env-entry-name>
      <env-entry-value>3600</env-entry-value>
      <env-entry-type>java.lang.Integer</env-entry-type>
   </env-entry>
…
```

The following settings apply to Web Application Descriptor for the Business Application Server:

Parameter	Context	Source
`fieldValuesAge`	How long the static cache is kept on the Web application server in seconds?	Defaults from template

Note: It is highly recommended that you do not change this configuration file by extracting the configuration file from the WAR/EAR file using java utilities, making the change manually and rebuilding the WAR/EAR. Use the initialSetup utility, with the –w option, to build the WAR/EAR file as documented in Business Application Server Configuration Process.

log4j.properties – Logging Configuration

Note: This log file should not be altered unless specified. The generated configuration file has all the recommended settings for all sites.

The product uses the *log4j* java classes to centralize all log formats into a standard format. The details of the configuration settings and *log4j* itself is available at

http://logging.apache.org/log4j/ or http://en.wikipedia.org/wiki/Log4j.

spl.properties – Product configuration settings

The product Business Application Server has a specific number of settings outside of the J2EE specification to control the internals of the product. This file exists as similar files exist for ALL modes of operation of the product (*for example, Batch can be run outside the J2EE Web application server*) so a common configuration standard was adopted.

For a description of all settings in the **spl.properties** file refer to the **spl.properties** parameter descriptions.

hibernate.properties – Database connectivity properties

Opening a connection to a database is generally much less expensive than executing an SQL statement. A connection pool is used to minimize the number of connections opened between application and database. It serves as a librarian, checking out connections to application code as needed. Much like a library, your application code needs to be strict about returning connections to the pool when complete, for if it does not do so, your application will run out of available connections. Hence, the need for having a connection pooling mechanism such as Hibernate using Oracle Universal Connection Pool (UCP) connection pooling or JNDI based connection pooling.

The online and Web Service components of the product use JNDI based connection pools and the batch component uses UCP based connection pools.

Hibernate is a powerful Object Relational Mapping (ORM) technology that makes it easy to work with relational databases. Hibernate makes it seem as if the database contains plain Java objects, without having to worry about how to get them out of (or back into) database tables. Coupled with the UCP or JNDI connection pooler, it provides a comprehensive connectivity tool for the COBOL/java to operate effectively against the database.

The product uses the Hibernate and either JNDI or UCP libraries to create a connection pool and connect the java/COBOL objects to the database to store, update, delete and retrieve data. It is used for all the database access for online as well as batch.

Refer to http://www.hibernate.org and http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/ucp.html for more information on the technology aspects of Hibernate and UCP.

The product has a configuration file for the database connectivity and pooling called the **hibernate.properties** configuration file. This file contains the configuration settings for the database connections and the connection pool to be used by any of the SQL statements accessing the database.

The configuration settings contained in the **hibernate.properties** file are summarized in the hibernate.properties section.

For a more indepth description of these parameters and others not included with the product see http://www.hibernate.org and http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/ucp.html .

Inbound Web Services Configuration

The Oracle Utilities Smart Grid Gateway product supports the ability to implement product Web Services within the J2EE Web Application Server instead of being accessible from the XML Application Integration (XAI) servlet. This has a number of advantages:

- Inbound Web Services can utilize the clustering and high availability features of the J2EE Web Application Server directly.
- Individual Inbound Web Services can be monitored from the J2EE Web Application Server directly. If Oracle Enterprise Manager is used, performance and diagnostic metrics on individual Inbound Web Services can be collected and analyzed for service level tracking and general monitoring.
- Individual Inbound Web Services can be secured using the WS-Policy support available in the J2EE Web Application Server directly. Customer using Oracle WebLogic Server can also utilize the inbuilt Oracle Web Services Manager within Oracle WebLogic to add access rules to individual Web Services.

Note: It is highly recommended to use Inbound Web Services instead of XML Application Integration as the XAI Servlet will be deprecated in a future release.

Enabling Inbound Web Services

By default, for backward compatibility, the Inbound Web Services feature is disabled. To use the feature it must be enabled and a few features enabled to use the deployment facilities provided by this capability. All these steps must be executed by the relevant administrator user as indicated in the individual steps.

Note: These steps, unless otherwise indicated, are performed once per environment.

Enable Inbound Web Services Feature

*Note: For customers using the native installation of the product, should ensure that the **WEB_WLS_SVRNAME** is set to the name of the server or cluster used for the environment.*

The first step is to enable the feature in the environment configuration file (**ENVIRON.INI**) to use additional facilities:

- Using the product administrator user on the machines where the product is installed, execute the **configureEnv[.sh] -a** command after executing the **splenviron[.sh] -e** *<environmentname>* where *<environmentname>* is the name of the installation environment to enable the feature upon.
- Change the option "*Enable Web Services Functionality*" on option 50 to **true**. Save the change using the "**P**" option.
- Execute the **initialSetup[.sh]** command to apply the changes to the infrastructure and generate the necessary build files for Inbound Web Services.

Set Credentials For Inbound Web Services

Note: Failure to perform this task may cause the deployment of Inbound Web Services to fail.

To deploy the Inbound Web Services seamlessly it is necessary to create an administration credential file so that the build and deployment utilities and online deployment facility for developers can access the domain.

To create this credential file the following commands must be executed:

- Using the product administrator user on the machines where the product is installed, execute the `splenviron[.sh] -e <environmentname>` where `<environmentname>` is the name of the installation environment to enable the feature upon.
- Execute the following command:

  UNIX:

  ```
  $JAVA_HOME/bin/java -classpath
  $WL_HOME/server/lib/weblogic.jar weblogic.Admin -username
  <username> -password <password> STOREUSERCONFIG -
  userconfigfile $SPLEBASE/etc/.wlsuserconfig -userkeyfile
  $SPLEBASE/etc/.wlsuserkey
  ```

  Windows:

  ```
  %JAVA_HOME%\bin\java -classpath
  %WL_HOME%\server\lib\weblogic.jar weblogic.Admin -username
  <username> -password <password> STOREUSERCONFIG -
  userconfigfile %SPLEBASE%\etc\.wlsuserconfig -userkeyfile
  %SPLEBASE%\etc\.wlsuserkey
  ```

  Where `<username>` is an Administration password for the domain and `<password>` is the password for that user. For example, these values can correspond to `WEB_WLSYSUSER` and `WEB_WLSYSPASS` (in plain text).

Note: If the password for this user is changed at anytime, this step must be repeated for the facility to continue to operate.

Enable USER_LOCK Facility

To prevent multiple parallel deployments, the `USER_LOCK` feature of the database must be enabled on the product user. The following commands must be executed:

- Logon onto the database using SQL Developer or SQL Plus the `SYS` or `SYSTEM` user.
- Execute the following command:

  ```
  @?/rdms/admin/userlock.sql
  ```

Note: ? corresponds to ORACLE_HOME.

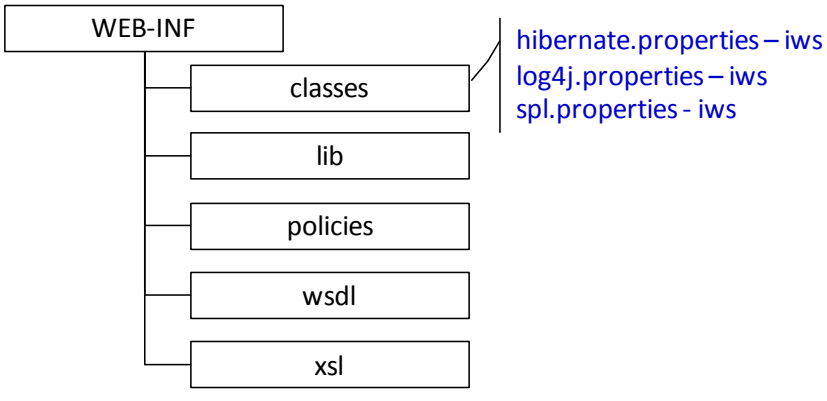
- Grant access to the product user using the following command:

  ```
  grant execute on USER_LOCK to <user>;
  ```

where ***<user>*** is the **DBUSER** and **BATCH_DBUSER**[2] from the **ENVIRON.INI**.

Inbound Web Services Concepts

The Inbound Web Services deploys a web archive (**webServices.war**) containing the components of the inbound Web Services using the following structures:



Structure	Contents
classes	Configuration Files for Inbound Web Services
lib	Product JAR Files
policies	WS-Policy files included natively in Inbound Web Services
wsdl	Generated WSDL Files
xsl	Style sheets to be used with Inbound Web Services

The Inbound Web Services was designed to fit within the same concepts as the Web Application Server/Business Application Services. The main differences are:

- The **webservices.war** file is located in the **$SPLEBASE/splapp/iws/gen** directory (or **%SPLEBASE%\splapp\iws\gen** on Windows) and is deployed automatically using iwsdeploy[.sh] or the online deployment facility.

  Note: Customers using native installation can optionally manually deploy the war file, if desired.

- Database connectivity is configured in the Inbound Web Services and included in the build file.

The rest of this section will outline the specific configuration and settings for the Inbound Web Services feature. Common settings across the architecture will be described in the Web Application Server and Business Application Server sections.

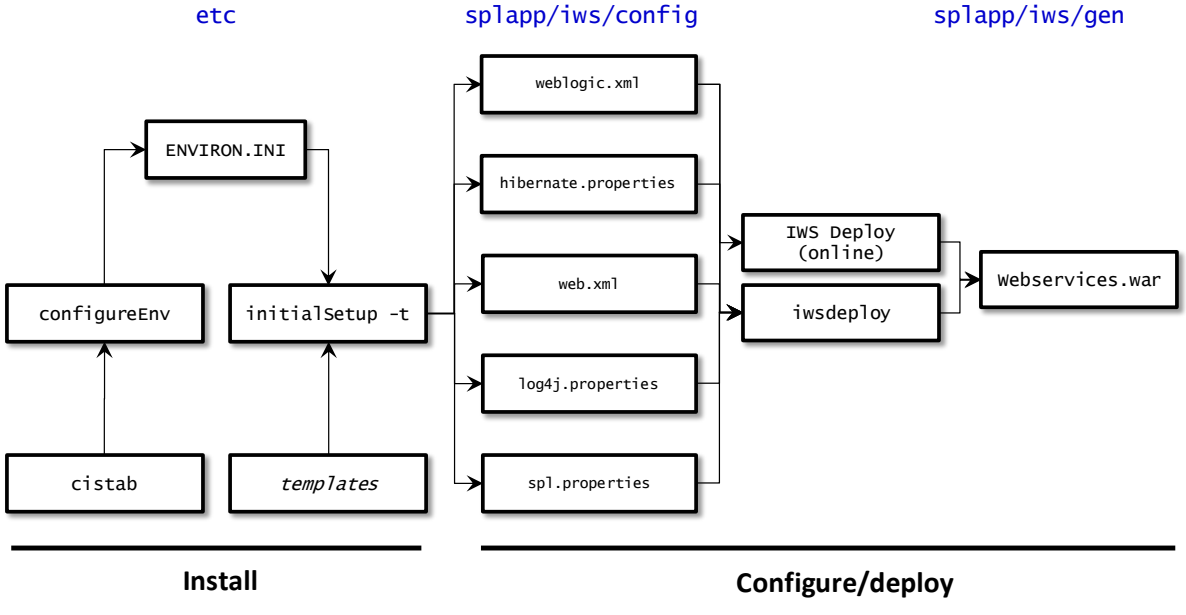
[2] **BATCH_DBUSER** is used for command line deployment.

Inbound Web Services Configuration Process

Note: This section only covers the configuration and deployment of the physical Web Service resources. It does not cover the definition of Inbound Web Services within the product.

Note: For this feature to operate, at least one Inbound Web Service should be defined for the build process to execute. If no Inbound Web Services are defined prior to the deployment then the build process will fail.

To configure the Inbound Web Services during the installation process and post-installation then the following process should be used:



- The configureEnv utility is used during installation time and can be used post implementation to set parameters in the ENVIRON.INI. If any parameters are derived or set from the ENVIRON.INI (see "Source" column in the relevant section) then the configureEnv utility should be used to maintain them.

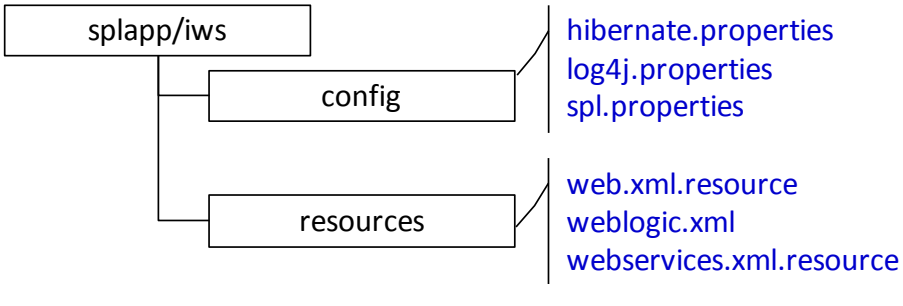
  Note: The configureEnv utility should be used to make ANY changes to the ENVIRON.INI. Manual changes to this configuration file are not recommended.

- After the ENVIRON.INI has been set or altered, the settings must be reflected in the relevant configuration files used by the Inbound Web Services by running the initialSetup -t:
  - log4j.properties – Logging Configuration
  - spl.properties – Product configuration settings
  - hibernate.properties – Database connectivity properties
- The utility uses the templates from the **templates** directory to create substituted copies of these files in a standard location:

Configuration File	Destination
Service Bean	

Configuration File	Destination
spl.properties Template: spl.properties.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/config Windows: %SPLEBASE%\splapp\iws\config
log4j.properties Template: log4j.properties.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/config Windows: %SPLEBASE%\splapp\iws\config
hibernate.properties Template: hibernate.properties.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/config Windows: %SPLEBASE%\splapp\iws\config
web.xml.resource Template: web.xml.resource.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/resources Windows: %SPLEBASE%\splapp\iws\resources
webservices.xml.resource Template: webservices.xml.resource.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/resources Windows: %SPLEBASE%\splapp\iws\resources
weblogic.xml Template: weblogic.xml.iws.template	Linux/Unix: $SPLEBASE/splapp/iws/resources Windows: %SPLEBASE%\splapp\iws\resources

The locations of the configuration files can be summarized in the following figure:



- At this point you may perform manual changes to the above files to parameters not implemented in the ENVIRON.INI.

Note: Any manual changes are overwritten after running the initialSetup utility unless the change is reflected in the appropriate template (see custom templates for more information). Backups should be made of any changes and then manually reapplied to reinstate all manual changes.

- To reflect configuration changes into the product the iwsdeploy utility, must be executed. This will build the necessary **spl-servicebean-<*version*>.jar** (where **<*version*>** is the version of the product used) and the **SPLService.ear** EAR file to be deployed into the J2EE Web application server. This step is optional if configuration overrides are in use (refer the discussion of allowing the externalization of configuration settings for alternative methods).

At this point the product Business Applications are ready for deployment into the J2EE Web application server.

Quick Reference Guide for Inbound Web Services Configuration

To make configuration changes to the IWS component of the product uses the following Quick Reference Guide to identify which process should be used:

- If the change is to any setting contained in the ENVIRON.INI for the IWS then you must run the following utilities in the order indicated:
  1. Execute the configureEnv utility to reflect the parameter change in the ENVIRON.INI.
  2. Execute the initialSetup utility (with the **–t** option) to rebuild the configuration files using the ENVIRON.INI and provided template files. This will reset the configuration to the contents of the base template files or custom template (if used.
  3. Any configuration changes that are overridden by templates (base or custom) must be manually reapplied (if necessary).
  4. Execute the initialSetup utility (with the **–w** option) to implement the configuration files in the product IWS files. This step is not necessary of you are using configuration overrides.
  5. Execute the iwsdeploy utility to build and deploy.
- If the change is to any setting not contained in the ENVIRON.INI for the IWS but is in the configuration files for the IWS then you must run the following utilities in the order indicated:
  1. Make any manual changes to the relevant configuration files.

Execute the initialSetup, with the **-w** option, utility to implement the configuration files in the product IWS files. This step is not necessary of you are using configuration overrides.

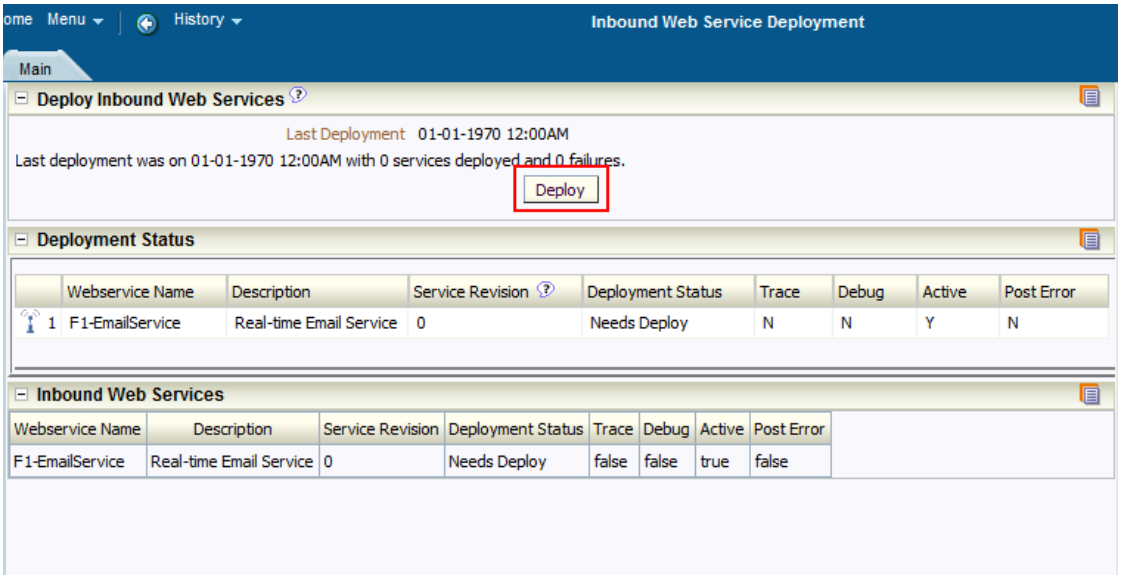
Inbound Web Services Deployment Process

After the configuration of the Inbound Web Services is complete (as outlined in Inbound Web Services Configuration Process) the final step to implement the product technically is to

deploy the product within the J2EE Web application server.

There are three methods of deploying the Inbound Web Services component within the J2EE Web application server:

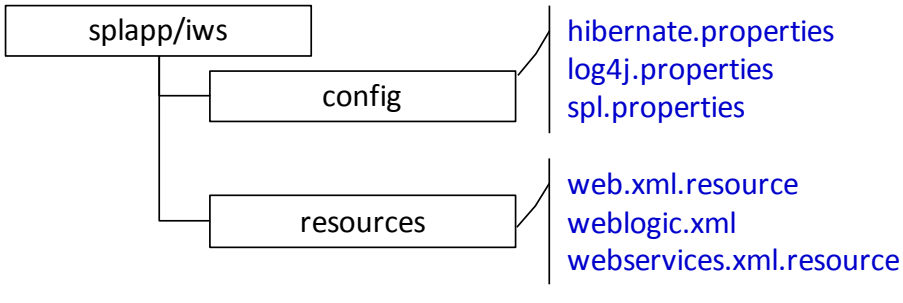
- An Inbound Web Services Deployment facility available from the product online. This is accessible from the *Admin → I → Inbound Web Services Deployment* menu item. This is suitable for development environments only. For example:



- Use the iwsdeploy utility to trigger a deployment from the command line.
- Execute the iwsdeploy utility but manually deploy the WebServices.war file to the server(s) or cluster using the J2EE console or deployment utilities.

Inbound Web Services Configuration Files

The following locations hold the Inbound Web Services configuration files:



The following configuration files dictate the Inbound Web Services configuration:

Location	Contents	Configuration Files
resources	J2EE Web Services Descriptor for Inbound Web Services	web.xml – J2EE Application Descriptor weblogic.xml - WebLogic Application Descriptor webservices.xml - Web

Location	Contents	Configuration Files
		Services Definition[3]
config	Application Configuration files for Inbound Web Serivces	log4j.properties – Logging Configuration hibernate.properties – Database connectivity properties spl.properties – Product configuration settings

[3] This file is fixed in format and should not be altered.

Miscellaneous Operations And Configuration

Enabling Email Logging from Log4j

The following sample configuration will enable email logging of ERROR level log messages in the product. When an error is encountered in startup and during operations of the product any ERROR message displayed on the console log file will be emailed to an Administrator's email account or email group.

Note: This change outlined below will make manual changes to a configuration file. Execution of initialSetup may overwrite these changes unless template overrides are used. Please ensure you make adequate backups to preserve this change. Refer to http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/net/SMTPAppender.html for details of the appender.

The following changes are required to enable this.

1) Open the log4j.properties in the relevant **$SPLEBASE/etc/conf** (**%SPLEBASE%\etc\conf** in Windows) subdirectory:
   - Web Application Server – log4j.properties
   - Business Application Server - log4j.properties
2) Add the following lines to the file:

```
###  E1 is an EmailAppender
log4j.appender.E1 = org.apache.log4j.net.SMTPAppender
log4j.appender.E1.Threshold = ERROR
log4j.appender.E1.layout = org.apache.log4j.PatternLayout
log4j.appender.E1.layout.ConversionPattern = %d{ISO8601} [%t] %-5p
%c %x - %m%n
log4j.appender.E1.From = <from>
log4j.appender.E1.SMTPHost = <SMTPHost>
log4j.appender.E1.Subject = <subject>
log4j.appender.E1.To = <to>
###
### The following settings are optional
###
log4j.appender.E1.SMTPUsername = <SMTPUsername>
log4j.appender.E1.SMTPPassword = <SMTPPassword>
log4j.appender.E1.CC = <cc>
log4j.appender.E1.BCC = <bcc>
```

Parameter	Field from example	Usage
From	*<from>*	Email address for emails
To	*<to>*	Email address/group to send emails to
CC	*<cc>*	Email address/group to send courtesy copy of emails to
BCC	*<bcc>*	Email address/group to send "blind" courtesy copy of emails to
SMTPHost	*<SMTPHost>*	Host Name of SMTP Server
SMTPUsername	*<SMTPUsername>*	Logon User for SMTP Server (*if supported*)
SMTPPassword	*<SMTPPassword>*	Password for Logon User for SMTP Server (*if supported*)
Subject	*<subject>*	Subject for email message

3) Modify the following lines in the log4j.properties file:

```
## System-wide settings
# set log levels - for more verbose logging change 'info' to
'debug' ###
log4j.rootCategory=info, A1, F1, E1
```

4) Execute the initialSetup utility, with the **–w** option, to reflect the changes in the WAR/EAR file.

5) To deploy the application refer to the Web Application Server Deployment Process or Business Application Server Deployment Process

Overriding the default Oracle database connection information

By default the database connection for Oracle databases is of the format:

jdbc:oracle:thin:@*<hostname>*:*<dbport>*:*<database_sid>*

where

<hostname>	Database hostname
<dbport>	Database Listener portname
<database_sid>	Database Name

The URL format is described at http://www.oracle.com/technology/tech/java/sqlj_jdbc/htdocs/jdbc_faq.html#05_03

This configuration setting is sufficient for the majority of the environments at a site. If your site requires a specialist URL for RAC support then you must override the default URL.

To override the default URL specify the following:

- Log on to the server containing the Business application server using the

administration account for the desired environment (for example, **splsys**).

- Execute the splenviron utility, with the **–e** option, to attach to the desired environment to change.
- Execute the configureEnv utility and choose to change menu block 4 (Database).
- Change the *Database Override Connection String* to the desired custom JDBC url.
- Press **p** to save the change to the ENVIRON.INI.
- Execute initialSetup, with the **–t** option, to reflect the change in the hibernate.properties files. This may overwrite custom changes if custom templates are not used.
- Execute initialSetup, with the **–w** option, to include the configuration changes in the WAR/EAR files. This option is not required if externalization of configuration is implemented.
- For selected platforms redeployment of the WAR/EAR files is required as per Business Application Server deployment process.

The following example uses the Oracle JDBC thin client (for Oracle Real Application Clustering):

```
jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = machine-name)(PORT = 1251))
    (ADDRESS = (PROTOCOL = TCP)(HOST = machine-name)(PORT = 1251)
    (LOAD_BALANCE = yes)
    (FAILOVER=YES)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SID.WORLD)
    )
)
```

Refer to Oracle RAC support for other examples.

Example URL using the Oracle JDBC thick client:

```
jdbc:oracle:oci:@SID.WORLD
```

Note: For thick client to work, the Oracle client library directory must be added to the library search path. Oracle client libraries are installed under ***ORACLE_HOME/lib*** *and* ***ORACLE_HOME/lib32*** *directories. Add this directory to the library search path environment variable. The library search path environment for AIX is* ***LIBPATH****, for HP-UX is* ***SH_LIB_PATH*** *for Linux is* ***LD_LIBRARY_PATH*** *and for Windows is* ***PATH****.*

Note: For Oracle Database 12.1 and above, use the format: ***jdbc:oracle:thin:@<hostname>:<dbport>:/<database_service>*** *to use Pluggable Databases where* ***<database service>*** *is the PDB name.*

Automatic shunning of Child COBOL JVM's

For products that use COBOL, there are a series of COBOL Child JVMs created for products

that support COBOL using the Oracle Utilities Application Framework for backward compatibility. This is primarily used to transfer data between the java based framework and any remaining COBOL based business objects.

There are instances when the COBOL processes hosted in *child* Java virtual machines can consume too many resources, e.g. running out of *native* memory. In the event that such a situation obtains, and cannot be resolved by e.g. identifying a problematic COBOL module, it is necessary to shutdown (*shun*) the OS process that hosts COBOL in order to reclaim the resources.

In these situations is possible to configure the system to automatically *shun* a COBOL child JVM in order to forestall a possible situation where the process consumes too many resources. This facility allows both time-based and request-based scheduling for an automated rollover to a standby JVM.

Optionally a facility has been created that allows for an automatic rollover from the active COBOL child JVM to a standby JVM, without disrupting any system processing. In order to allow this, the system must be configured to use at least two (2) child JVMs, to assure a near-instantaneous switchover to the standby JVM.

The feature is activated by placing either, or both, of the following properties into the spl.properties that govern the Child JVM:

```
spl.runtime.cobol.remote.jvmMaxLifetimeSecs=[number of seconds]
spl.runtime.cobol.remote.jvmMaxRequests=[number of COBOL requests]
```

Set either property to zero (or leave it out) to disable the relevant rollover policy.

- If the JVM max lifetime seconds parameter is set to e.g. 3600 for one hour, then one hour after the first request is made to that child JVM, it will be automatically *shunned*, completing all in-flight requests normally, while transferring all new work to the standby child JVM.
- If the JVM max requests parameter is set to e.g. 50000, then after 50000 COBOL commands have been sent to the child JVM, it will be automatically *shunned* as above.
- When both parameters are provided, the child JVM will be shunned automatically when either condition obtains, e.g. shun after one hour, or 20000 COBOL commands, whichever comes first

Note: These policies are not active in the default configuration as part of the installation process there must be manually added to online spl.properties files or added to a custom template version of spl.properties.services.template.

The system creates log file entries when a rollover condition has been satisfied.

Cache Management

A great deal of information in the system changes infrequently. In order to avoid accessing the database every time this type of information is required by an end-user, the system maintains a cache of static information on the Web Application Server. In addition to the Web Application Server cache, information is also cached on each client browser.

Server Cache

Note: Maintenance of the cache is performed automatically by the product. Whilst there are commands to force refreshes of the cache, these are designed for administrator and developer use only. Additional security setup is required to enable individual users to access to the facilities below.

The cache is populated the first time any user accesses a page that contains cached information. For example, consider a control table whose contents appear in a dropdown on various pages. When a user opens one of these pages, the system verifies that the list of records exists in the cache. If so, it uses the values in the cache. If not, it accesses the database to retrieve the records and saves them in the cache. In other words, the records for this control table are put into the cache the first time they are used by any user. The next user who opens one of these pages will have the records for this control table retrieved from the cache (thus obviating the database access).

Typically, this information

The following points describe the type of data that is cached on the web server:

- **Field labels.** This portion of the cache contains the labels that prefix fields on the various pages in the system.
- **System information.** This portion of the cache contains installation and license key information as well as basic information about the various application services (e.g., the URL's that are associated with the various pages).
- **Menu items.** This portion of the cache contains the menu items.
- **Dropdown contents.** This portion of the cache contains the contents of the various dropdowns that appear throughout the system.
- **XSL documents.** This portion of the cache contains each page's static HTML.
- **Portal information.** This portion of the cache contains information about which zones are shown on the various pages.

The contents of the cache are cleared whenever the Web Application Server is restarted or as automatically refreshed as controlled by the `fieldValuesAge` parameter on the Web Application Server web.xml configuration file. This means that fresh values are retrieved from the database upon first use by end users.

If you change the database after the cache is built and the information you changed is kept in the cache, users may continue to see the old values. If you don't want to restart your Web Application Server, you can either use the relevant operation on the JMX FlushBean Mbean available on the Web Application Server or issue a custom browser URL to issue the appropriate command (see below).

Note: To use the browser URL for the resetting of the cache the user must be logged on to the product browser interface and have access to the `F1ADMIN` application service.

Function	JSP	MBean Operation
Refresh all cache	`flushAll.jsp`	`flushAll`

Function	JSP	MBean Operation
Refresh all drop down data	flushDropdownCache.jsp	flushDropDownCache
Refresh field labels	flushMessageCatalog.jsp	flushMessageCatalog
Refresh Fields and FK information	flushFieldAndFKMetaData.jsp	flushFieldAndFKMetaData
Refresh menu items	flushMenu.jsp	flushMenu
Refresh messages	flushMessaging.jsp	flushMessaging
Refresh navigation keys	flushNavigationInfo.jsp	flushNavigationInfo
Refresh portals and zones	flushPortalMetaInfo.jsp	flushPortalMetaInfo
Refresh screen style sheets	flushUI_XSLs.jsp	flushUIXSLs
Refresh security	flushSystemLoginInfo.jsp	flushSystemLoginInfo
Refresh specific drop down data	flushDropDownField.jsp	flushDropDownField

Note: It is recommended that the "Refresh all cache" is used for non-production and production systems. The other commands are designed for primarily for development use only. Refer to the Oracle Utilities SDK documentation for more information about the options available with the commands.

Note: When using these commands the cache will be reloaded over time with fresh data. As the data is loaded there is a negligible delay in each transaction that reloads data into the cache for the first time. Therefore it is recommended not to execute this command frequently.

Client Cache

In addition to the server cache, information is cached on each user browser. After clearing the cache that's maintained on the Web Application Server, it is recommended to also clear the cache that is maintained on the client browser (if possible). To do this, follow the following steps:

Browser	Steps
Microsoft Internet Explorer	• Select *Tools* on your browser menu bar • Select *Internet Options* on the menu that appears. • Click the *Delete Files* button on the pop-up that appears. • Click the *Delete all…* button on the subsequent pop-up that appears and then click OK. • Enter the standard product URL to re-invoke the product.
Mozilla Firefox	• Select *Tools* from your browser menu bar. • Click *Options* on the Tools menu.

Browser	Steps
	• Select the *Advanced* tab from the Options dialog. • Select the *Network* tab from the Advanced tab. • Click on the *Clear Now* button. • Enter the standard product URL to re-invoke the product.

Note: Each user's cache is automatically refreshed as controlled by the ***maxAge*** *and* ***maxAgeI*** *parameters in the Web Application Server web.xml configuration file. . We recommend that you set these parameter to 1 second on development / test environments and 28800 seconds (8 hours) on production environments.*

Oracle WebLogic: Expanded or Archive Format

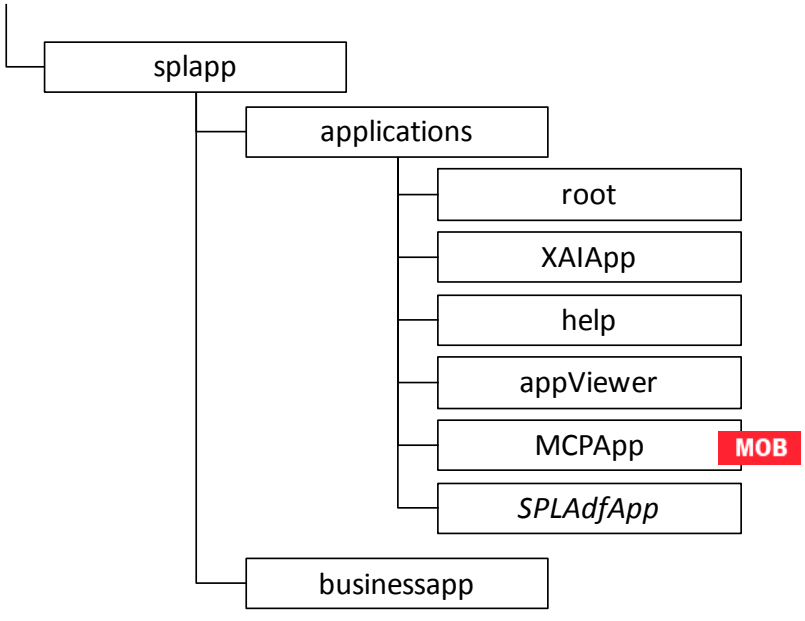
Note: Expanded format is not support on IBM WebSphere or IBM WebSphere ND.

By default, the product is built into a set of WAR/EAR files and deployed in this format on Oracle WebLogic and IBM WebSphere/ND to operate. For Oracle WebLogic it is possible to use *expanded* mode rather than the WAR/EAR format. This mode allows the Oracle WebLogic instance directories access to the directories and files used by the J2EE components of the product without the need for WAR/EAR files. This has a number of key advantages:

- Changes to the individual files in the product (such as JSP's or graphics) do not require a rebuild of the WAR/EAR file.
- Outage time to deploy and execute the WAR/EAR file is reduced as Oracle WebLogic reads the files directly. In the deployment process, Oracle WebLogic loads the WAR/EAR file and uncompressed it to a staging or temporary location for actual execution. This is greatly reduced under *expanded* mode as the files are already uncompressed.
- Application of patches and service packs is faster as the patch installer does not need to rebuild the WAR/EAR files after applying patches.

This expanded mode is suggested for non-production and demonstration environments and is not recommended for production (the default is *Archive* [non-expanded] mode) as the during the WAR/EAR process additional integrity checks are performed and security control of individual application files adds higher security requirements to production.

The figure below illustrates the expanded mode main directories:



- Under the **root** directory are the product specific subdirectories for each subsystem or part of the online component of the product.
- Under the **XAIApp** directory are the product specific subdirectories for each subsystem or part of the Web Services component of the product.
- Under the **busienssapp** directory are the business object specific files for each subsystem or part of the online component of the product.
- The **help** and **AppViewer** directories contain an expanded mode version of the **help** HTML (and related files) and **appViewer** generated files (after running genappvieweritems).
- Under the **MCPApp** directory are the Mobile Communications Platform specific files (if the mobile framework is deployed with your product). MOB
- Under the **SPLAdfApp** directory are any ADF specific files if the Oracle ADF interface is installed and used with your product.

Implementing Custom Templates

As described in the Web Application Server Configuration Process and Business Application Server Configuration Process the configuration files used in the product are built from templates. These templates are typically located in the **$SPLEBASE/templates** (or **%SPLEBASE%\templates** on Windows) subdirectory of each environment.

*Note: The file **FW_template_structure.xml** in the **structures** subdirectory lists all the templates and their destination paths. This file should not be altered unless instructed by Oracle Support.*

By default the product uses the base produce provided templates to build the configuration files. These configuration files are usually adequate for most needs in non-production but usually require some customization for production or site specific standards not covered by the base templates. In the past the site had two options:

- **Make custom changes to the configuration file directly** – This can performed against the **$SPLEBASE/etc/conf** (**%SPLEBASE%\etc\conf** on Windows) copies of the configuration files. The issue here is that if the configuration files are reset back to the templates intentionally or unintentionally, using the initialSetup utility, custom manual changes may be lost if not reapplied manually.
- **Make custom changes to base configuration templates** – In extreme conditions it was possible to make manual changes to the base product templates to reflect your site standards and customizations. The issue is that new releases of the templates for new features would overwrite any customizations if not reapplied manually.

To address this it is now possible to override base product templates with a copy of the template (a custom template). This can be achieved by copying the desired base template in the templates directory to the same name prefixed with "**cm.**". The initialSetup utility will use the custom template instead of the base template.

The process to implement this is as follows:

- Identify the template in the that is used by the desired configuration file. Use the information in the Web Application Server Configuration Process and Business Application Server Configuration Process sections of this document to help identify the templates used for each configuration file.
- Copy the desired template in the **$SPLEBASE/templates** (or **%SPLEBASE%\templates** on Windows) subdirectory to the same name but prefixed with a "**cm.**". This will be the override custom template. To disable the custom template at any time either rename the template to another name or remove it from the subdirectory.
- Make the necessary adjustments to the custom template as per your site standards. Please follow any conventions used in the template including use of environment variables or configuration settings from ENVIRON.INI.
- Use initialSetup as per Web Application Server Configuration Process and Business Application Server Configuration Process sections of this document to use the template to generate the new configuration files and incorporate the changes in the product.

Note: If custom templates are implemented, it is the sites responsibility to maintain the custom templates to reflect any changes in the base templates for new, changed or removed functionality.

Additional templates

The templates mentioned in previously in this document are the main configuration file based templates. There are additional configuration files that are built and used for various purposes. Most of these configuration files are used internally for management of the infrastructure and generation of utilities.

*Note: The file **FW_template_structure.xml** in the **structures** subdirectory lists all the templates and their destination paths. This file should not be altered unless instructed by Oracle Support.*

There are a number of areas the templates cover:

- **Configuration Files for Oracle WebLogic** – Oracle WebLogic has specific requirements for configuration settings and files. Refer to Oracle WebLogic Configuration Support for more specific details.
- **Configuration Files for other software** – Third party software has specific requirements for configuration files.
- **Utilities for deployment** – Additional configuration files are built to use in the deployment process to define the product applications to the relevant runtime software.
- **Internal ANT build configuration files** – Configuration and build files are built to support the configuration build process.

Note: The latter two categories of templates and configurations (utilities and ANT build files) should not be altered unless instructed by Oracle Support.

The table below lists the templates in the template directory not covered by other sections of this document applicable to the online, service and XAI components:

Templates	Configuration File	Usage
`application.xml.template`	`applicaton_web.xml`	J2EE global application configuration file, which contains common settings for the Web Application Server
`application_service.xml.template`	`application_service.xml`	J2EE global application configuration file, which contains common settings for the Business Application Server
`billdirfile.ini.template`	`billdirfile.ini`	Bill Print extract configuration file
`boot.properties.template`	`boot.properties`	Oracle WebLogic boot credentials file used for starting server
`coherence-cache-config.xml.template`	`coherence-cache-config.xml`	Batch Coherence cache settings. BATCH
`config.xml.template` `config.xml.win.template`	`config.xml`	Oracle WebLogic main configuration file. The `win.template` is used for the Windows environments.
`doc1dirfile.ini.template`	`doc1dirfile.ini`	Bill Print extract configuration file

Templates	Configuration File	Usage
earServiceBuild.xml.template	earServiceBuild.xml	ANT Build file for EAR file for Business Application Server
earWebBuild.xml.template	earWebBuild.xml	ANT Build file for EAR file for Web Application Server
ejb-jar.xml.template	ejb-jar.xml	Generic Business Application Server descriptor for EJB's
ibm-application-bnd.xmi.template	ibm-application-bnd.xmi	Deployment descriptor for IBM WebSphere/ND.
jarservice.xml.template	jarservice.xml	ANT Build file for jar files.
java.login.config.template	java.login.config	JAAS Login file used for XAI servlet. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
jps-config.xml.template	jps-config.xml	ADF security configuration.
MPLIsUp.cmd.template	MPLIsUp.cmd	Utility to check status of MPL (if used) as called by spl[.sh] on Windows. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
MPLIsUp.sh.template	MPLIsUp.sh	Utility to check status of MPL (if used) as called by spl[.sh] on Linux/UNIX. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
MPLParameterInfo.xml.template	MPLParameterInfo.xml	MPL Configuration file. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My

Templates	Configuration File	Usage
		Oracle Support for more details.
ouaf.jmx.access.file.template	ouaf.jmx.access.file	Default access file for JMX.
ouaf.jmx.password.file.template	ouaf.jmx.password.file	Default security file for JMX.
OUAF-Target.xml.template	OUAF-Target.xml	Oracle Identity Manager interface configuration File. Refer to My Oracle Support KB Id **970785.1** for details of this integration.
setDomainEnv.cmd.template	setDomainEnv.cmd	Utility to set Domain configuration for Oracle WebLogic on Windows.
setEnv.sh.template	setEnv.sh	Utility to set Oracle WebLogic environment variables.
splcobjrun.cmd.template	splcobjrun.cmd	COBOL runtime command (if COBOL used) for Windows.
splcobjrun.sh.template	splcobjrun.sh	COBOL runtime command (if COBOL used) for Linux/Unix.
startMPL.cmd.template	startMPL.cmd	Utility to start MPL (if used) as called by spl[.sh] on Windows. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
startMPL.sh.template	startMPL.sh	Utility to start MPL (if used) as called by spl[.sh] on Linux/UNIX. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
startWebLogic.cmd.template	startWebLogic.cmd	Utility to start Oracle WebLogic on Windows.

Templates	Configuration File	Usage
startWebLogic.sh.template	startWebLogic.sh	Utility to start Oracle WebLogic on Linux/UNIX.
startWLS.sh.template	startWLS.sh	Utility invoking JVM for Oracle WebLogic.
stopMPL.cmd.template	stopMPL.cmd	Utility to stop MPL (if used) as called by spl[.sh] on Windows. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
stopMPL.sh.template	stopMPL.sh	Utility to stop MPL (if used) as called by spl[.sh] on Linux/UNIX. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more details.
stopWebLogic.cmd.template	stopWebLogic.cmd	Utility to stop Oracle WebLogic on Windows.
system-jazn-data.xml.template	system-jazn-data.xml	ADF security store definitions.
tangasol-coherence-override.xml.template	tangasol-coherence-override.xml	Batch Coherence overrides BATCH
warbuild.xml.template	warbuild.xml	ANT WAR Build file
warupdate.xml.template	warupdate.xml	ANT WAR file for updates
weblogic.policy.template	weblogic.policy	Java Security file used by Oracle WebLogic to protect the product files.
weblogic-ejb-jar.xml.template	weblogic-ejb-jar.xml	Deployment descriptor for Business Application Server for Oracle WebLogic.
XAIParameterInfo.xml.template	XAIParameterInfo.xml	XAI Configuration file. Refer to XAI Best Practices whitepaper KB Id: **942074.1** on My Oracle Support for more

Templates	Configuration File	Usage
		details.

Note: Templates not mentioned in this document that exist in the templates directory are included in one or more templates above depending on the configuration requirements. Templates relating to the Batch component of the architecture are covered in the Batch Server Administration Guide.

Oracle WebLogic Configuration Support

Whilst the product supports multiple J2EE Web Application Server vendors, the product has *native* support for Oracle WebLogic. Normally the J2EE Web Application is installed and the J2EE Web Application Server components are embedded in the directories controlled by the Web Application Server software during the deployment process. The deployment process usually transfers the WAR/EAR files to the J2EE Web Application Server directories (varies according to J2EE Web Application Server software).

For Oracle WebLogic, the Oracle WebLogic software is effectively *pointed* to directories as in the product installation. This avoids Oracle WebLogic having additional copies of its configuration and WAR/EAR files under its own directory structure.

In this case the following configuration aspects of Oracle WebLogic apply:

- The **`$SPLEBASE/splapp`** (or **`%SPLEBASE%\splapp`** in Windows) subdirectory is referenced directly in the configuration files.
- In non-expanded mode (see Oracle WebLogic: Expanded or Archive Format for details), the WAR/EAR files are directly referenced from the config.xml file.
- In expanded mode (see Oracle WebLogic: Expanded or Archive Format for details), the application files are directly reference in the **`splapp`** subdirectories from the config.xml file.
- The config.xml file is located under **`splapp/config`** rather than using the Oracle WebLogic location. Any changes made from the Oracle WebLogic console are stored in this file.
- The utilities to start and stop the Oracle WebLogic instance are located under the **`splapp`** subdirectory.
- The security configuration files for the Oracle WebLogic instance are located under the **`splapp`** subdirectory. The security repository configured is configured in the location supplied with the Oracle WebLogic instance.

Thus facility allows one installation of Oracle WebLogic to be used across many environments with each environment being independent.

Using Configuration Files outside the WAR/EAR file

Typically, the configuration files specified Web Application Server Configuration Process and Business Application Server Configuration Process are embedded into the WAR/EAR files, as per the J2EE specification, ready for deployment for use at runtime. While this is

generally acceptable for most sites, it also means that any configuration change requires rebuilding of the WAR/EAR files and redeployment to fully implement the configuration changes. This may add outage time to implement configuration changes.

It is possible to allow the product to use versions of the certain configuration files outside the WAR/EAR files to minimize outage time to implement changes. In most cases, a restart of the product components is necessary to implement the configuration change.

The table below outlines the configuration files that can be *externalized* from the WAR/EAR file by product component:

Component	Configuration File	Externalized
Web Application Server (root and XAIApp)	web.xml	✕
	spl.properties	✓
	weblogic.xml	✕
	log4j.properties	✓
Business Application Server	web.xml	✕
	spl.properties	✓
	hibernate.properties	✓
	log4j.properties	✓

By default, the externalization works on the following principles:

- The **SPLEBASE** environment variable must be set to the home location of the software prior to execution of the Web Application Server or Business Application Server. This must match the value configured for the environment in the cistab configuration file on the machine.
- The external versions of the configuration files should be in their default locations (as supplied) in the **$SPLEBASE/etc/conf** (or **%SPLEBASE%\etc\conf** for Windows) subdirectories.
- The product use the external configuration file versions instead of the versions embedded in the WAR/EAR files. If you wish to revert to the embedded versions then the site can either rename the **conf** subdirectories to prevent the external configuration files being detected or ensuring the **SPLEBASE** environment is not set.

*Warning: If the **conf** subdirectories are renamed they should be reverted to their original names before ANY single fix, service pack or upgrade is performed to prevent configuration reset to base templates or installation failure.*

This facility is useful for a number of situations:

- If any passwords are changed that are used by the product on a regular basis, reflecting changes in the configuration files directly or using templates is easier using externalized configuration files. The WAR/EAR files do not need to be rebuilt and redeployed and this can save time.
- During the initial phases of production or when traffic volumes fluctuate, it may be

necessary to tune specific settings. This allows experimentation of the changes before committing to specific values. It allows greater level of *flexibility* in configuration change.

Note: It is recommended to ensure that in the long term that both the external versions and embedded versions are kept in synch on a regular basis to prevent configuration issues. This can be done using standard maintenance windows as necessary.

Oracle RAC Support

Note: Refer to the Oracle Real Application Clustering (RAC) documentation for setup instructions and parameter settings for RAC. It is assumed that RAC is installed, including Oracle Notification Service (ONS) for Fast Connection Failover support and configured prior to configuration of the product to take advantage of the RAC installation.

The product supports the use of Oracle's Real Application Clustering (RAC) for high availability and performance through database clustering. The product has additional setting to tell the database pooling aspects of the product to take advantage of the RAC facilities.

Once RAC has been installed and configured on the database there are a number of options that can be used to configure the product to use RAC in all modes of configuration:

- It is possible to setup a custom DB Connection string to take advantage of the RAC as outlined in the Overriding the default Oracle database connection information section of this document. This is the easiest implementation of RAC but does not take advantage of the full RAC features.
- Configure RAC specific settings in the installation configuration files (via the `configureEnv[.sh]` utility). The following settings should be set:

Environment Setting	Usage	Comments
`ONS_JAR_DIR`	Location of ONS Jar file (`ons.jar`)	This is the location of the Oracle Notification Service Jar files for use in the product.
`ONSCONFIG`	ONS configuration string with RAC server nodes delimited by "`,`" in the form *`<host>:<port>`* where *`<host>`* is the RAC host node and *`<port>`* is the ONS listener port.	Used for connections

Note: Native RAC Support does not support XA transactions using Universal Connection Pool (UCP) at the present time. If XA compliance is required, it is suggested that JNDI based pools provided by the Web Application server be used as documented in Using JNDI Based Data Sources.

Note: At the present time Oracle Single Client Access Name (SCAN) is not supported in the configuration of RAC native support.

Note: Once configured the ***spl.runtime.options.isFCFEnabled*** *is set to true and* ***spl.runtime.options.onsserver*** *is set to the value specified in* ***ONSCONFIG****.*

Note: Support for Implicit Connection Caching has been removed as this feature has been superseded by Universal Connection Pool (UCP).

Adding a custom Privacy policy screen

In certain sites the product must display a privacy policy to remind users of privacy rules at a site. The product allows for a custom HTML based page to be added by the site. The privacy page should be named **privacy.html** and placed in the **cm** directory so that the URL is:

Error! Hyperlink reference not valid.

where

<host>	Host Name of the Web Application Server used by the product
<port>	Port Number allocated to the Web Application Server used by the product
<server>	Server context allocated to Web Application Server used by the product

Refer to the Oracle Utilities SDK on how to add custom HTML to the product.

Once implemented the privacy statement can be obtained from the above URL or the following URL:

http://*<host>*:*<port>*/*<server>*/privacy

where

<host>	Host Name of the Web Application Server used by the product
<port>	Port Number allocated to the Web Application Server used by the product
<server>	Server context allocated to Web Application Server used by the product

IBM WebSphere/WebSphere ND Support

Whilst the product supports both Oracle WebLogic and IBM WebSphere there are specific additional options available for IBM WebSphere and IBM WebSphere ND. The list below summarizes the specific additional support for these Web Application Servers:

- The application within IBM WebSphere is set to the following values:

Tier	Usage
Web Application Server	**SPLWeb-*<WEB_SVRNAME>*** where ***<WEB_SVRNAME>*** is the value of the **WEB_SVRNAME** environment setting.
Business Application Server	**SPLService-*<BSN_SVRNAME>*** where ***<BSN_SVRNAME>*** is the value of the **BSN_SVRNAME** environment setting.

- The following IBM WebSphere specific environment settings (**ENVIRON.INI**) should

be specified for correct basic operation:

Environment Setting	IBM WebSphere edition
BSN_APP (SPLService)	WAS WASND
BSN_NODENAME	WASND
BSN_SRVNAME	WAS WASND
BSN_WLHOST	WAS WASND
WAS_HOME	WAS
WASND_DMGR_HOST	WASND
WASND_HOME	WASND
WEB_APP (SPLWeb)	WAS WASND
WEB_NODENAME	WASND
WEB_SVRNAME	WAS WASND
WEB_WLHOST	WAS WASND

These variables are used by the initialSetup utility to build and deploy the EAR/WAR files correctly.

- A number of Python scripts are used by the utilities to interface to IBM WebSphere administration API:

Command Script	Usage
webspherеDeployService.py	Deploy Business Application Service WAS
webspherеDeployWeb.py	Deploy Web Application Service WAS
webspherеNDDeployService.py	Deploy Business Application Service WASND
webspherеNDDeployWeb.py	Deploy Web Application Service WASND
webspherеNDStartService.py	Start Business Application Service WASND
webspherеNDStartWeb.py	Start Web Application Service WASND
webspherеNDStopService.py	Stop Business Application Service WASND
webspherеNDStopWeb.py	Stop Web Application Service WASND
webspherеNDUnDeployService.py	Undeploy Business Application Service WASND
webspherеNDUnDeployWeb.py	Undeploy Web Application Service WASND
webspherеStartService.py	Start Business Application Service WAS
webspherеStartWeb.py	Start Web Application Service WAS

Command Script	Usage
`websphereStopService.py`	Stop Business Application Service WAS
`websphereStopWeb.py`	Stop Web Application Service WAS
`websphereUnDeployService.py`	Undeploy Business Application Service WAS
`websphereUnDeployWeb.py`	Undeploy Web Application Service WAS

- The utilities to deploy/undeploy (`initialSetup`) the Web and Business Application WAR/EAR files and start/stop the server (`spl`) utilize the IBM WebSphere `wsadmin` command. Refer to the IBM WebSphere/ND documentation for more details of this command.
- The `initialSetup` utility, provided with the product, operate at the node level and not the cluster level for IBM WebSphere/ND. Customers wanting to deploy/undeploy at the cluster level should use the `wasadmin` command natively or use the IBM WebSphere administration console to achieve this.

User Exit Include Files

Whilst the product supports custom templates it is now possible to only supply fragments of a customization rather than whole configuration templates, known as *user exit include files*. This allows you to specify additional settings to be included in the templates provided *in stream* when the product templates are used to generate the configuration files when using the `initialSetup` command.

When `initialSetup` is executed the templates are applied with the following order of preference:

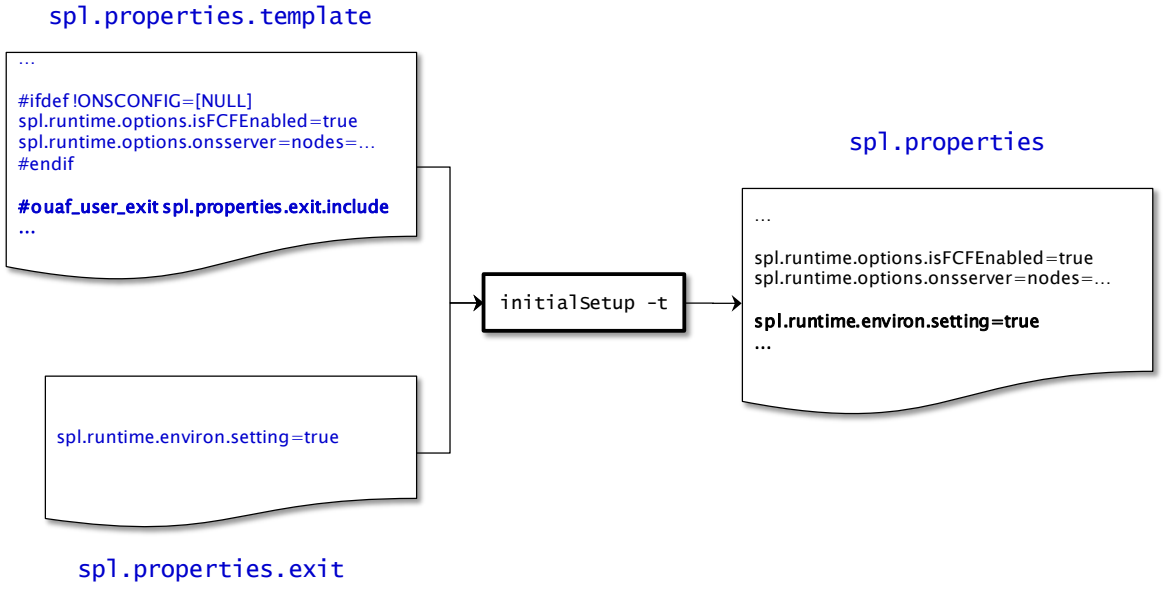
- Base framework templates (no prefix). These templates should not be altered.
- If a product specific template exists (prefixed by the product code) then the product template is used instead of the base Framework template for the configuration file. These templates should not be altered.
- If a template is prefixed with "`cm_`" then this is a custom template to be used instead of the product specific and base framework template.

These templates should live in `$SPLEBASE/templates` (or `%SPLEBASE%\templates` on Windows).

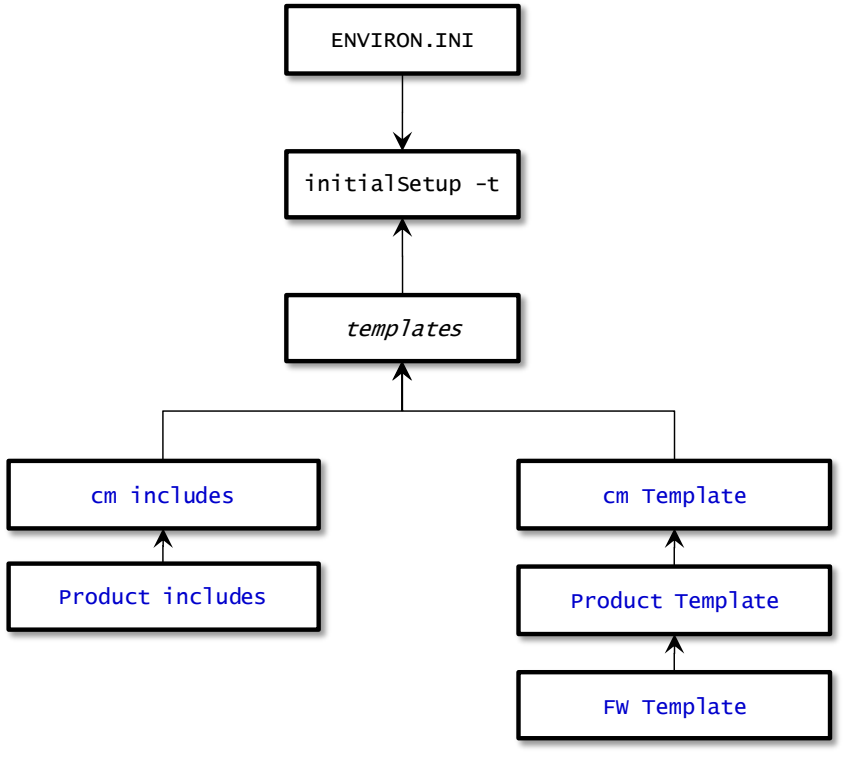
Note: When creating custom templates please use the base framework and any related product templates as the basis for the content of the custom template.

Whilst this facility is flexible it means that any updates to the base or product templates MUST be reflected in any custom templates. A new option is to use user exits that are placed strategically in the most common configuration files that need change. When `initialSetup` is executed the existence of user exit files are checked (when an `#ouaf_user_exit` directive exists in the template) and the contents included in the

generated configuration file. The figure illustrates the process for a typical configuration change:



As with the custom templates user exits have preferences depending on the ownership of the user exit include file. Custom includes will override any product specific includes. There are no base includes as they are already included in the template files. The figure below illustrates the preferences for both templates and includes:



The table below outlines the currently available user exits in the available templates:

Template File	User Exit Include file	Position and Usage
boot.properties.template	boot.properties.exit.include	Used to change boot properties file from

Template File	User Exit Include file	Position and Usage
		Oracle WebLogic
config.xml.template[4]	config.xml.exit_1.include	Before ADF deployment information (used for ADF)
	config.xml.exit_2.include	After Web Application container definition
	config.xml.exit_3.include	End of Configuration File
	config.xml.exit_4.include	Before Web Application container definition
	config.xml.jms.include	WebLogic JMS Configuration
config.xml.win.template[5]	config.xml.win.exit_1.include	Before ADF deployment information (used for ADF)
	config.xml.win.exit_2.include	After Web Application container definition
	config.xml.win.exit_3.include	End of Configuration File
	config.xml.win.exit_4.include	Before Web Application container definition
	config.xml.win.jms.include	WebLogic JMS Configuration
ejb-jar.xml.template	ejb-jar.xml.wls.jms_1.include	JMS Mappings for Oracle WebLogic
	ejb-jar.xml.was.jms_1.include	JMS Mappings for IBM WebSphere/ND
hibernate.properties.web.template	hibernate.properties.exit.include	At end of file (common hibernate.properties entries)
	hibernate.properties.web.exit.include	At end of file (online specific hibernate.properties entries)
log4j.properties.template	log4j.properties.exit.include	At end of file (common

[4] This is the template for the Oracle WebLogic instance, refer to the Oracle WebLogic for an example of contents.

[5] This is the template for the Windows version of Oracle WebLogic.

Template File	User Exit Include file	Position and Usage
		log4j.properties entries)
	log4j.properties.root.exit.include	At end of file (specific online log4j.properties entries)
log4j.properties.XAIApp.template	log4j.properties.exit.include	At end of file (common log4j.properties entries)
	log4j.properties.XAIApp.exit.include	At end of file (specific XAI log4j.properties entries)
log4j.properties.service.template	log4j.properties.exit.include	At end of file (common log4j.properties entries)
	log4j.properties.service.exit.include	At end of file (specific XAI log4j.properties entries)
ouaf.jmx.access.file.template	ouaf.jmx.access.file.exit.include	Allows for additional users to be specified for JMX connections
ouaf.jmx.password.file.template	ouaf.jmx.password.file.exit.include	Allows for additional passwords to be specified for JMX users
splcobjrun.cmd.template	splcobjrun.cmd.exit.include	Allows for COBOL execution parameters (COBOL supported products only) - Windows
splcobjrun.sh.template	splcobjrun.sh.exit.include	Allows for COBOL execution parameters (COBOL supported products only) – Linux/UNIX
spl.properties.service.template	spl.properties.exit.include	At end of file (common spl.properties entries)
	spl.properties.service.exit.include	At end of file for EJB spl.properties entries.
	spl.properties.service.timeouts.exit.include	User exit for service timeouts.
spl.properties.template	spl.properties.exit.include	At end of file (common spl.properties

Template File	User Exit Include file	Position and Usage
		entries)
	spl.properties.root.exit.include	At end of file for Web Application based spl.properties entries.
	spl.properties.timeouts.root.exit.include	User exit for global timeouts
spl.properties.XAIApp.template	spl.properties.exit.include	At end of file (common spl.properties entries)
	spl.properties.XAIApp.exit.include	At end of file for XAI Application based spl.properties entries.
	spl.properties.XAIApp.timeouts.exit.include	Future use
web.xml.template	spl.properties.images.include	Image processing overrides for web.xml
	web.xml.servlet_mapping.include	Allow custom servlet mappings
	web.xml.servlet.include	Allow custom servlet definitions
	spl.properties.filter_mapping.include	Allow custom filter mappings

To use these user exits create the user exit include file with the prefix "cm_" in the $SPLEBASE/templates (or %SPLEBASE%\templates) directory. To reflect the user exits in the configuration files you must execute the initialSetup utility. Refer to the Custom JMS Configuration section for an example of this process.

Properties File User Exits

The product behavior is controlled at a technical level by the values in the properties files. Whilst most of the settings are defaulted to their correct settings in the file, additional parameters may be added to the properties files to add new behavior. User exits are used to set these additional parameters in the properties files.

From the table above there are more than one user exit available in each properties file template to use. This is designed to maximize the reusability of configuration settings. There are a number of specialized user exits that may need to be used:

- **Common Settings** – The configuration files used by each channel of execution (online, Web Services and batch) has a common user exit. This user exit is used to

house all the setting you want to implement regardless of the channel used. For example the common setting user exits are:

Configuration File	User Exits for common settings
hibernate.properties	hibernate.properties.exit.include
log4j.properties	log4j.properties.exit.include
spl.properties	spl.properties.exit.include

- **Channel specific Settings** – To implement custom settings per channel there is a separate user exit to hold those parameters for those channels. The specific user exits are:

Channel	Configuration File	User Exits for common settings
Web App Server	hibernate.properties	hibernate.properties.web.exit.include
	log4j.properties	log4j.properties.root.exit.include
	spl.properties	spl.properties.root.exit.include
Business App Server	log4j.properties	log4j.properties.service.exit.include
	spl.properties	spl.properties.service.exit.include
Web Services	log4j.properties	log4j.properties.XAIApp.exit.include
	spl.properties	spl.properties.XAIApp.exit.include

Custom JMS configuration

The product includes a realtime Java Message Services (JMS) connector to provide application to application integration. To use this facility the physical JMS definitions need to be defined as part of the configuration to be included in the configuration of the J2EE Web Application Server[6]. These will match the JMS configuration within the product itself. Refer to the installation documentation provided with the product to understand the required JMS integration.

To reflect the JMS settings a number of custom user exist include files have been provided to implement the JMS changes in the `config.xml` and `ejb-jar.xml` configuration files required by the J2EE Web Application Servers.

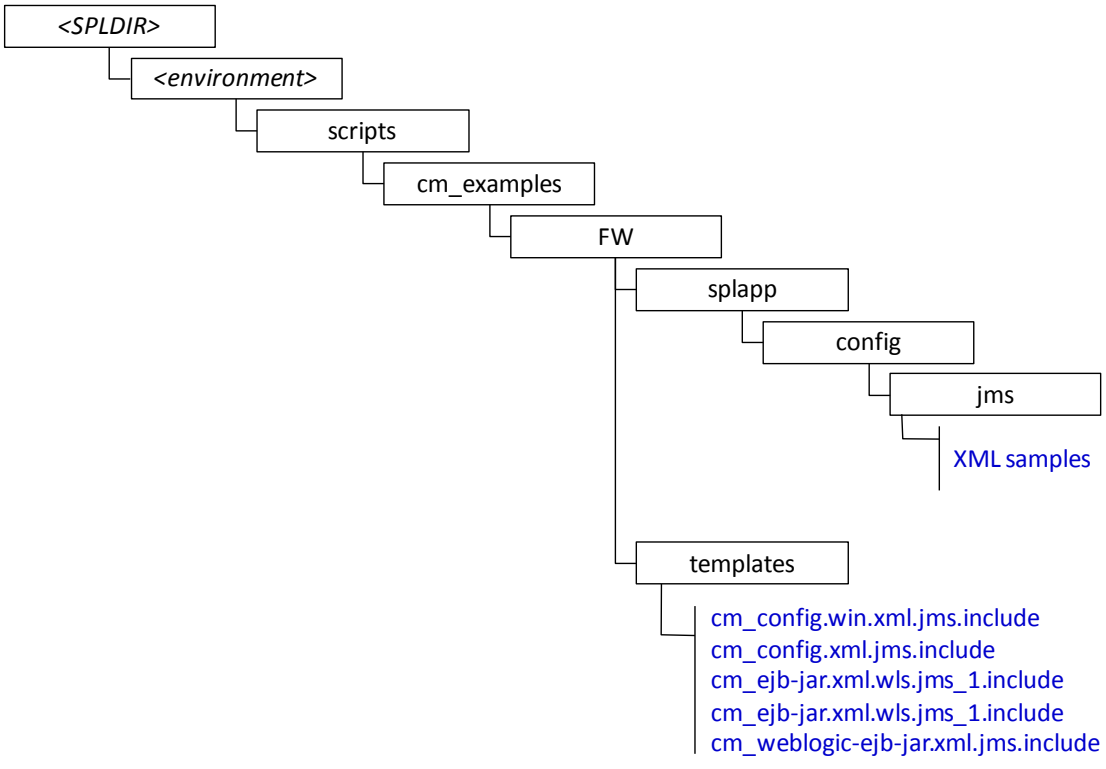
There are two sets of files that are provided for the JMS integration as examples:

- A set of user exit include files in the `scripts/cm_examples/FW/templates` directory for the various files necessary to define the physical JMS configuration settings.
- A set of sample XML files that define the attributes of sample JMS settings that are

[6] It is possible to define the physical JMS configuration using the console provided with the J2EE Web Application Server but this may be overwritten during upgrades.

referred to by the custom templates user exit include files. These are the files that need to be maintained by the site according to product or local standards.

The figure below illustrates the location of the sample JMS files:



To implement the JMS configuration files at your site:

- Logon to the Web Application Server and/or Business Application Server machine using the product administration account.
- Execute the **splenviron** command to orientate to the desired environment you wish to apply the JMS configuration to.
- Create a JMS configuration repository directory under **/splapp/config/jms**. The following:

  Linux/Unix:

  ```
  mkdir -p $SPLEBASE/splapp/config/jms
  ```

  Windows:

  ```
  mkdir %SPLEBASE%\splapp\config\jms
  ```

- Copy the sample XML configuration files to the directory created using the following commands:

  Linux/Unix:

  ```
  cp $SPLEBASE/scripts/cm_examples/FW/splapp/config/jms/*
  $SPLEBASE/splapp/config/jms/
  ```

  Windows:

  ```
  xcopy %SPLEBASE%\scripts\cm_examples\FW\splapp\config\jms\*
  %SPLEBASE%\splapp\config\jms\
  ```

- Copy the user exit include files to the template directory so that the user exits are

implemented whenever **initialSetup** is executed.

Linux/Unix:

```
cp $SPLEBASE/scripts/cm_examples/FW/templates/*
$SPLEBASE/templates/
```

Windows:

```
xcopy %SPLEBASE%\scripts\cm_examples\FW\templates\*
%SPLEBASE%\templates
```

*Note: To reverse out the change at any time the template files that are copied (prefixed by cm) can be removed from the templates subdirectory under **%SPLEBASE%** or **$SPLEBASE**.*

- Modify the sample XML files in the **config/jms** directory to suit your product requirements or your site requirements.

At configuration time the settings provided these files are included in the target configuration files as indicated by the user exit include files.

Online Transaction Timeouts

By default the product does not impose any transaction time limits on online and web services transactions. If this is not appropriate for your site then transaction time limits can be implemented globally as well as on individual objects using configuration settings defining the desired transaction time limits.

Specific user exits should be used to maintain these settings. To implement these create or modify the user exit files indicated in the table below in **$SPLEBASE/templates** (or **%SPLEBASE%\templates** on Windows).

To impose global limits the following settings must be added to your user exit files:

Tier/configuration file	Configuration Setting
Web Application Server (**cm_spl.properties.timeouts.root.exit.include** user exit file)	• Set the **ouaf.timeout.query.default** parameter to the desired timeout (in seconds) to set a global default on query zones.
Business Application Server (**cm_spl.properties.service.timeouts.exit.include** user exit file)	• Set the **ouaf.timeout.business_service.default** parameter to the desired timeout (in seconds) to set a global default on business service invocations. • Set the **ouaf.timeout.business_object.default** parameter to the desired timeout (in seconds) to set a global default on business object invocations. • Set the

Tier/configuration file	Configuration Setting
	ouaf.timeout.script.default parameter to the desired timeout (in seconds) to set a global default on service script invocations. • Set the **ouaf.timeout.service.default** parameter to the desired timeout (in seconds) to set a global default on application service invocations.

To impose timeout values on individual object/service/scripts then an entry in the Business Application Server **cm_spl.properties.service.timeouts.exit.include** user exit file must exist for each individual object/service/script to specify the timeout:

Object	Comments
Business Objects	Default: ouaf.timeout.business_object.default Specific: ouaf.timeout.business_object.*<bocode>*
Business Services	Default: ouaf.timeout.business_service.default Specific: ouaf.timeout.business_service.*<bscode>*
Query Zones	Default: ouaf.timeout.query.default Specific: ouaf.timeout.query.*<zonename>*
Scripts	Default: ouaf.timeout.script.default Specific: ouaf.timeout.script.*<scriptname>*
Application Services	Default: ouaf.timeout.service.default Specific: ouaf.timeout.service.*<service>*

For example:

ouaf.timeout.service.CILTUSEP=600

Note: Timeout values are not precise as they do not include additional time needed to process any rollback or networking activity necessary after a timeout has occurred.

Note: Timeout user exits exist for batch and XAI as well but they are not used in the current release of the product. These are reserved for potential use in future releases.

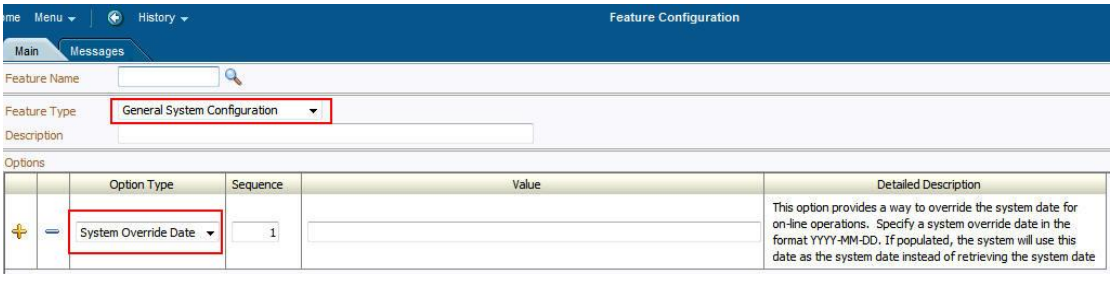
Setting the Date for testing purposes

One of the common techniques used in testing is to set the date to a fixed point in time to simulate data aging in the product. By default, the date (and time) used in the system is obtained from the database server with the time zone used on the user record to offset (if used by the product). It is possible to override the system date used at a global level or at an individual user level for testing purposes.

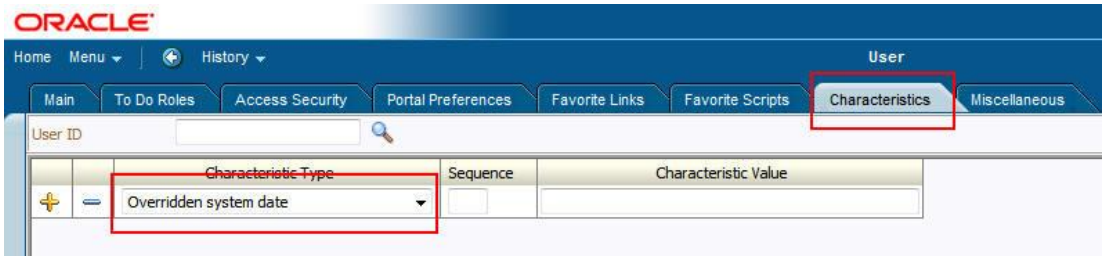
Note: This facility is not recommended for use in Production environments.

To use this facility the following must be configured:

- Set the **`spl.runtime.options.allowSystemDateOverride`** to **`true`** in the **`spl.properties`** file for the online (Web or Business Application Server), XAI (Business Application Server) and/or Batch (standalone).
- To set the feature at the global level for an environment, navigate to the *Administration* menu → *F* → *Feature Configuration* menu option and add a *General System Configuration* Feature Type with the *System Override Date* option in **`YYYY-MM-DD`** format. For example:



- If individual test users wish to use different dates they can set the feature at the user level. This feature does not require the global override setting to be used but if the global override is also used, then this user specific setting is used. To set the date override at the user level, add the *Overridden system date* Characteristic Type to the individual user record (via *Administration* menu → *U* → *User* menu option) with the Characteristic Value set to the desired date in **`YYYY-MM-DD`** format. For example:



When the facility is active the following applies:

- The facility will only be active if the **`spl.runtime.options.allowSystemDateOverride`** parameter is set to **`true`** in the relevant **`spl.properties`** file.

- If the system override via Feature configuration is set then all users using that channel will use that date for any transactions. If the Feature configuration is not set then the default current date as per the database server is used.
- If users have system date overrides as characteristics then that user for that channel will use that date regardless if the system override is set or not.

Simple Web Application Server context

By default, the Web Application server will directly connect to the Database to load its cache at startup time. Customers, who implement the product installation in distributed mode, where the Web Application Server and Business Application Server are deployed separately, may wish to prevent the Web Application Server to connect to the database directly.

In the Advanced Web Application Server configuration, it is possible to set the *Create Simple Web Application Contex*t to **true** to force the Web Application Server to load its cache via the Business Application rather than direct loading.

When setting this value to **true**, the following properties files should be manually removed prior to executing the product:

$SPLEBASE/etc/conf/root/WEB-INF/classes/hibernate.properties

$SPLEBASE/splapp/applications/root/WEB-INF/classes/hibernate.properties

*Note: For customers who are using a local installation, where the Web Application Server and Business Application Server are combined in the deployed server, should set this parameter to **false**, the default, unless otherwise required.*

Secure Transactions

The product supports HTTP and HTTPS protocols for transmission of data from the browser client and within the architecture. Customers must choose either HTTP (unsecure) or HTTPS (secured) for protocol. Use of both protocols simultaneously is not supported. The default protocol is HTTP.

If you wish to implement HTTPS protocol then the following process must be used:

- The value for **WEB_WLSSLPORT** must be specified for the SSL port to use. When this is specified then HTTP is disabled automatically.
- The product ships with the demonstration certificate shipped with the Web Application Server software. It is not recommended to use this certificate for your site. It is highly recommended that you obtain a certificate for your site from a trusted source and install the certificate as per the Web Application Server documentation.
- For all traffic directly to the product please use the **https** protocol on the URL's used for direct interaction (via the browser or Web Services interfaces).

Note: For Oracle WebLogic customers, refer to the Configuring Identity And Trust section of the Oracle WebLogic Installation Guide.

Note: For both protocols, the PUT, DELETE, TRACE and OPTIONS methods not permitted in the security constraints for the product by default.

Killing Stuck Child JVM's

Note: This facility is only applicable to products using COBOL based extensions.

In some situations, the Child JVM's may *spin*. This causes multiple startup/shutdown Child JVM messages to be displayed and recursive child JVM's to be initiated and shunned. If the following:

Unable to establish connection on port …. after waiting .. seconds.

The issue can be caused intermittently by CPU spins in connection to the creation of new processes, specifically Child JVMs. Recursive (or double) invocation of the `System.exit` call in the remote JVM may be caused by a `Process.destroy` call that the parent JVM always issues when shunning a JVM. The issue may happen when the thread in the parent JVM that is responsible for the recycling gets *stuck* and it affects all child JVMs.

If this issue occurs at your site then there are a number of options to address the issue:

- Configure an Operating System level kill command to force the Child JVM to be shunned when it becomes stuck.
- Configure a `Process.destroy` command to be used if the kill command is not configured or desired.
- Specify a time tolerance to detect stuck threads before issuing the `Process.destroy` or `kill` commands.

Note: This facility is also used when the Parent JVM is also shutdown to ensure no zombie Child JVM's exit.

The following additional settings must be added to the spl.properties for the Business Application Server to use this facility:

- `spl.runtime.cobol.remote.kill.command` – Specify the command to kill the Child JVM process. This can be a command or specify a script to execute to provide additional information. The kill command property can accept two arguments, `{pid}` and `{jvmNumber}`, in the specified string. The arguments must be enclosed in curly braces as shown here.

  Note: The PID will be appended to the killcmd string, unless the {pid} and {jvmNumber} arguments are specified. The jvmNumber can be useful if passed to a script for logging purposes.

  Note: If a script is used it must be in the path and be executable by the OS user running the system.

- `spl.runtime.cobol.remote.destroy.enabled` – Specify whether to use the `Process.destroy` command instead of the kill command. Specify `true` or `false`. Default value is `false`.
- `spl.runtime.cobol.remote.kill.delaysecs` – Specify the number of

seconds to wait for the Child JVM to terminate naturally before issuing the **Process.destroy** or **kill** commands. Default is 10 seconds.

For example:

spl.runtime.cobol.remote.kill.command=kill -9 {pid} {jvmNumber}

spl.runtime.cobol.remote.destroy.enabled=false

spl.runtime.cobol.remote.kill.delaysecs=10

When a Child JVM is to be recycled, these properties are inspected and the **spl.runtime.cobol.remote.kill.command**, executed if provided. This is done after waiting for **spl.runtime.cobol.remote.kill.delaysecs** seconds to give the JVM time to shut itself down. The **spl.runtime.cobol.remote.destroy.enabled** property must be set to **true** AND the **spl.runtime.cobol.remote.kill.command** omitted for the old *Process.destroy* command to be used on the process.

Note: By default the spl.runtime.cobol.remote.destroy enabled is set to false and is therefore disabled.

If neither **spl.runtime.cobol.remote.kill.command** nor **spl.runtime.cobol.remote.destroy.enabled** is specified, child JVMs will not be forcibly killed. They will be left to shut themselves down (which may lead to orphan JVMs). If both are specified, the **spl.runtime.cobol.remote.kill.command** is preferred and **spl.runtime.cobol.remote.destroy.enabled** defaulted to **false**.

It is recommended to invoke a script to issue the direct kill command instead of directly using the *kill -9* commands.

For example, the following sample script ensures that the process Id is an active cobjrun process before issuing the kill command:

forcequit.sh

```
#!/bin/sh
THETIME=`date +"%Y-%m-%d %H:%M:%S"`
if [ "$1" = "" ]
then
   echo "$THETIME: Process Id is required" >>$SPLSYSTEMLOGS/forcequit.log
   exit 1
fi
javaexec=cobjrun
ps e $1 | grep -c $javaexec
if [ $? = 0 ]
then
   echo "$THETIME: Process $1 is an active $javaexec process -- issuing
kill
-9 $1" >>$SPLSYSTEMLOGS/forcequit.log
  kill -9 $1
  exit 0
else
  echo "$THETIME: Process id $1 is not a $javaexec process or not active --
  kill will not be issued" >>$SPLSYSTEMLOGS/forcequit.log
```

```
  exit 1
fi
```

This script's name would then be specified as the value for the **spl.runtime.cobol.remote.kill.command** property, e.g:

```
spl.runtime.cobol.remote.kill.command=forcequit.sh
```

The **forcequit** script does not have any explicit parameters but pid is passed automatically.

To use the jvmNumber parameter it must explicitly specified in the command. For example, to call script **forcequit.sh** and pass it the pid and the child JVM number, specify it as follows:

```
spl.runtime.cobol.remote.kill.command=forcequit.sh {pid} {jvmNumber}
```

The script can then use the JVM number for logging purposes or to further ensure that the correct pid is being killed.

If the arguments are omitted, the pid is automatically appended to the **spl.runtime.cobol.remote.kill.command** string.

Using Oracle Enterprise Manager

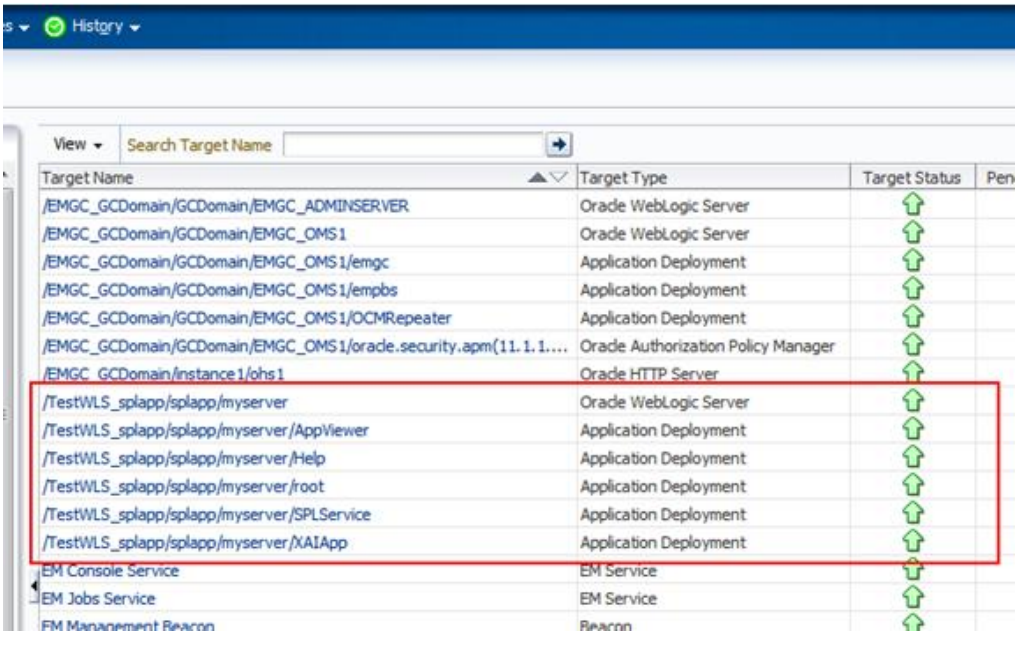
Oracle Enterprise Manager can discover and manage the products using the Oracle Application Management Pack for Oracle Utilities.

It is possible to manage and monitor the database and Oracle WebLogic from Oracle Enterprise Manager. When using native mode, Oracle Enterprise Manager will autodiscover the Oracle WebLogic instance using its native facilities. To use Oracle Enterprise Manager with environments using the default embedded support of Oracle WebLogic the following can be used to discover and monitor the instance:

- Within Oracle Enterprise Manager console, navigate to the *Add Targets Manually* menu option under the Setup menu.
- Select *Add Non-host Targets using Guided Process* from the options list.
- Select *Oracle Fusion Middleware* to denote that Oracle WebLogic will be discovered.
- In the dialog specify the following values:
    - **Administration Server Host** - The host name used for **WL_HOST** in your environment. This host must be registered to Oracle Enterprise Manager as a target so that the agent is redeployed.
    - **Port** - The port number assigned to the environment (**WL_PORT**).
    - **Username** - An account authorized to the Oracle WebLogic console. The Oracle Utilities Application Framework installer creates an initial user system that can be used if you have not got a site specific value for this user. This userid is used, by default, for all operations to the target. It must be an Administration account not a product account.
    - **Password** - The password configured for the Username.

- **Unique Domain Identifier** - An unique identifier for the domain to denote within Oracle Enterprise Manager. This is important and should be some value that means something for your administrator to understand. This also allows multiple targets per host to be defined easily. Make sure you do not use any embedded blanks and special characters for the name.
- **Agent** - This is the default host and port for the OEM agent on that machine. Just for references and can be altered if the default port is different for OEM at your site.

- Choose to *Continue* and the above target will be registered for use within Oracle Enterprise Manager.
- Each server in your domain will be registered as an *Oracle WebLogic Server* and every component of the product will be registered as an *Application Deployment*. For example:



Native Oracle WebLogic Support

One of the features of the product is the ability to use the Oracle WebLogic features in either embedded or native mode. In non-production it is recommended to use embedded mode unless otherwise required. Customers using Oracle ExaLogic for non-production should use native mode to fully support Oracle ExaLogic's architecture.

Whilst all the details of installing the product in native mode is covered in the Installation Guide a summary of what is required is shown below:

- A copy of the Oracle WebLogic must be installed on the machine. This copy of Oracle WebLogic must not be shared across multiple environments. Using native mode restricts a single copy of the product to an individual installation of Oracle WebLogic. Customers requiring multiple environments on a single installation should use embedded mode or install multiple Oracle WebLogic installation and use Oracle Enterprise Manager to manage the multiple instances.

- When using native mode, the product installation should not be placed under a users home directory or under the Oracle WebLogic home location. It should be installed in a separate location and using the deployment utilities deployed into the Oracle WebLogic domain location.
- Oracle WebLogic must be setup and configured with the following before deployment is to be performed:

Configuration Setting	Comments
Domain should be created	The Oracle WebLogic domain to install the product upon should be created with the Administration Server active on that environment.
Servers should be created	Using the Oracle WebLogic console the Servers to house the product should be created.
Create XML Registry	Using the Oracle WebLogic console an XML Registry to define the default parser should be created. On AIX this is done at the Oracle WebLogic command line level. Refer to the Installation Guide for more details.
Set Java parameters in console	Set the Domain level java settings for memory etc as per the Installation Guide.
Define Security	Define the Security Role, Security Realm and other Security definitions for the product as per the Installation Guide.
Create SYSUSER	Create the initial User for the product (SYSUSER) and attach the security role created earlier.
Set **SPLEBASE** variable	Prior to deployment and execution ensure the **SPLEBASE** variable is set to point to the location of the product as per the Installation Guide

- To start and stop the online component of the product, in *native* mode, it is recommended to use the facilities provided by Oracle WebLogic. This can be either using the Oracle WebLogic console, Oracle WebLogic utilities or via Oracle Enterprise Manager.
- To monitor the online component of the product use the facilities provided in Oracle WebLogic console, Oracle WebLogic utilities or via Oracle Enterprise Manager. Additional monitoring capabilities are available using the *Oracle Application Management Pack for Oracle Utilities*.
- When making changes to the product anytime the EAR files are changed they must be redeployed using the Oracle WebLogic console.

Redeploying Web Services

Note: This facility is only available for Oracle WebLogic.

After an XAI Inbound Service is defined it must be registered with the server to allow

Oracle WebLogic Console Operations

Note: Parts of this section do not apply to implementations using the embedded mode. Refer to individual sections for guidance.

When using the native mode of Oracle WebLogic to manage the product, the Oracle WebLogic administration console can be used to perform common operations. This section outlines the common operations that can be used from the console. Refer to the Administration documentation supplied with Oracle WebLogic for further advice.

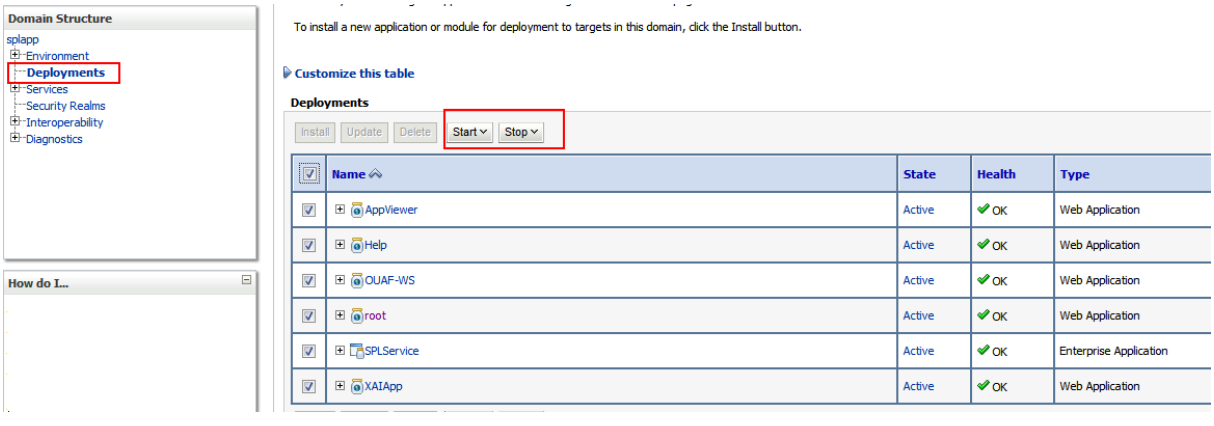
Starting and Stopping from console

Note: This facility should only be used for customers using native mode. Customers using embedded mode should use the spl utility.

As the product is using the native mode and native utilities provided by Oracle WebLogic it is possible to manage the product start and stop from the console. To perform this function, the user must do the following:

- Login to the console designated to administrate the environment using an appropriate administration account[7].
- Select the *Deployments* section of the *Domain Structure*. This will list the deployments to the domain.
- Select the appropriate deployments to start or stop.
- Use the *Start* or *Stop* function to perform the start or stop operation, respectively.

For example:



While it is possible to start or stop individually components it is recommended that all components be started at once to avoid availability issues. If individual components are

[7] The default installation account is `system` if used.

started or stopped use the following guidelines to minimize issues:

- Be aware of the architecture when stopping components. When starting the product, components lower in the architecture must be started first. When stopping the product, components higher in the architecture should be stopped first. Use the table below as a guide:

Deployment	Role
root	Web Application Server
SPLService	Business Application Server
XAIApp/OUAF-WS	XML Application Integration
Help	Online Help
AppViewer	Data Dictionary

- Stopping non-essential runtime components such as Online Help and AppViewer may result in HTTP 404 error when users select these functions unless they are sourced from other servers.

Monitoring Web Applications using the console

Note: This facility is available regardless of mode (i.e. embedded or native) used on the environment.

The Oracle WebLogic administration console can be used to monitor the state and performance of the individual components of the product. This can be achieved using the *Monitoring* tab of the *Deployments* option of the *Domain Structure* section of the Oracle WebLogic administration console.

This allows the following to be monitored:

- **Web Applications** – Statistics relating to the Web Applications deployed to the server. The Context Root and Source Information outlines the individual component monitored.
- **JMS** – If you are using the Message Driven Bean functionality within the product to integrate to the JMS resources in the Oracle WebLogic server then statistics related to this function can be monitored.
- **EJBs** – The Business Application Server and Message Driven Bean functionality are expressed as Enterprise Java Beans (EJB) and can be monitored from the console. The EJB SPLServiceBean is the Business Application server.
- **Web Services/Web Service Clients** – When using the native Web Service capability the statistics applicable to individual Web Services can be monitored.
- **JDBC** – If JDBC data sources are manually configured then statistics relating to these connections can be monitored.
- **Workload** – By default, Oracle WebLogic allocates simple workload managers for each deployment. These workload managers and any custom constraints can be monitored.

Refer to the Oracle WebLogic documentation for a detailed description of each of the statistics and their relevance.

JDBC Support

By default, the product uses the Universal Connection Pooling (UCP) to manage database connections. It is also possible to use Oracle WebLogic connection pooling via Data Sources for the online transactions (UCP will continued to be used for batch transactions). To use Data Sources the data sources must be created within Oracle WebLogic and then configuration files altered to utilize the Data Source using the following process:

- Create the Data Source in the Oracle WebLogic console. Specify the following:

Attribute	Comment
Name	Allocate a name for monitoring and management purposes
JNDI Name	Allocate a name to be used by the connection. This is used as *<datasourcename>* later.
Database Type	Specify `Oracle`
Database Driver	Specify the database driver to use. *Do not use the XA versions of the driver for the product.* By default use the *Thin driver for Service Connections*.
Supports Global Transactions	Deselect this option as it does not apply.
One Phase Commit	Ensure this option is selected.
Database Name	Specify the Oracle SID of the database to connect to. This should correspond to **DBNAME** in the **ENVIRON.INI**.
Host Name	Specify the host allocated to Oracle. This should correspond to **DBSERVER** in the **ENVIRON.INI**.
Port	Specify the Oracle Listener Port number of the database to connect to. This should correspond to **DBPORT** in the **ENVIRON.INI**.
Database User Name	Specify the product database user. This should correspond to **DBUSER** in the **ENVIRON.INI**.
Password	Specify the password for the user. This should correspond to **DBPASS** in the **ENVIRON.INI**.
Initial Capacity	This is the initial size of the connection pool
Maximum Capacity	This is the maximum size of the connection pool
Capacity Increment	This is the number of connections that are added to the connection pool when necessary.

Attribute	Comment
Statement Cache Type	This denotes the type of cache to use. Use the **LRU** setting.
Statement Cache Size	This is the cache size.

- Using a telnet terminal logon to the central location for the software using the product administration account.
- Execute the **splenviron** utility to set the environment variables for the environment.
- Navigate to the **$SPLEBASE/templates** folder.
- Copy the **hibernate.properties.web.template** file to **cm_hibernate.properties.web.template** file. This creates a custom template.
- Edit the **cm_hibernate.properties.web.template** file and replace the following content:
    - Remove the **hibernate.connection.url** lines from the file.
    - Remove all UCP parameters (**hibernate.ucp.*** parameters).
    - Add the following lines:

      ```
      hibernate.connection.datasource=<datasourcename>
      hibernate.connection.username=<JNDI Datasource user name>
      hibernate.connection.password=<JNDI Datasource password>
      ```

    where

<datasourcename>	JNDI Name
<JNDI Datasource user name>	User used to access the JNDI
<JNDI Datasource password>	Password of user to access JNDI

- Save the file and execute the **initialSetup** utility to reflect the change.
- If using native mode, redeploy the product EAR files.

Appendices - Parameters

Conventions in this section

This section contains a number of conventions to illustrate attributes and scope of configuration parameters:

- Valid values are indicated for each parameter and where applicable the default which is indicated in **bold**.
- The scope of the parameter is indicated with relevant icons:

WEB	Web Application Server
BAS	Business Application Server
IWS	Inbound Web Services
XAI	XML Application Integration
DB	Database
BATCH	Background Processing
MOB	Mobile based products only

- Some parameters depend on the J2EE Web Application Server used. The following icons will be used to denote the applicable J2EE Web Application Server. No icon indicates the parameter applies across all supported J2EE Web Application Server:

WLS	Oracle WebLogic
WAS	IBM WebSphere
WASND	IBM WebSphere Network Deployment

- Some parameters apply to embedded mode installation settings for Oracle WebLogic. These do not need to be set for native mode installations. The following icons will indicate these settings:

EMBED	Embedded Mode
NATIVE	Native Mode

- The ordinality of the parameter is indicated. Parameters that are mandatory must exist and are set in the relevant configuration file for the product to operate.
- Where applicable an example is indicated with the format.

ENVIRON.INI

The **ENVIRON.INI** is the environment file that is output from **configureEnv** and is used by **initialSetup** to build the configuration files from templates.

ADDITIONAL_RUNTIME_CLASSPATH - Additional Classpath

Parameter	ADDITIONAL_RUNTIME_CLASSPATH
Description	Additional Runtime Classpath for Web Application Server to allow custom jar libraries to be added to path, for Oracle WebLogic. For native mode customers, additional jar libraries can be added on the classpath on the Server definition.
Ordinality	Optional
Valid Values	Valid Classpath
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

ADDITIONAL_STOP_WEBLOGIC - Additional Stop Arguments

Parameter	ADDITIONAL_STOP_WEBLOGIC
Description	Additional Stop arguments when running the Administration Server on a different port. Refer to Starting and Stopping Servers for more information.
Ordinality	Optional
Valid Values	Valid Stop options for Oracle WebLogic
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

ADF_HOME - Location of ADF Software

Parameter	ADF_HOME
Description	Location of Oracle ADF libraries. Only set if ADF Zone Type is used in implementation or within product.
Ordinality	Optional
Valid Values	Full path to ADF libraries on server
J2EE Web Application Server	WLS
Tier	WEB BAS
Installation Mode	EMBED NATIVE

ANT_ADDITIONAL_OPT - Additional options for ANT

Parameter	ANT_ADDITIONAL_OPT
Description	Implementation specific options for WAR and EAR builds used by initialSetup.
Ordinality	Optional
Valid Values	Valid ANT options
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

ANT_HOME - Location of ANT

Parameter	ANT_HOME
Description	Generated location of ANT used by initialSetup. This value is autogenerated and should not be altered.
Ordinality	Mandatory
Valid Values	Valid location of ANT
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

ANT_OPT_MAX - ANT Maximum Heap Size

Parameter	ANT_OPT_MAX
Description	Maximum memory in MB allocated to ANT builds. Unless otherwise instructed by Oracle Support this value should not be altered.
Ordinality	Optional
Valid Values	Default: **800**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

ANT_OPT_MIN - ANT Minimum Heap Size

Parameter	ANT_OPT_MIN
Description	Minimum memory in MB allocated to ANT builds. Unless

	otherwise instructed by Oracle Support this value should not be altered.
Ordinality	Optional
Valid Values	Default: **200**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

appViewer - Name of appViewer WAR file

Parameter	appViewer
Description	Name of AppViewer WAR file
Ordinality	Mandatory
Valid Values	This value is automatically generated
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

BATCHDAEMON - Whether Online Batch Daemon is enabled

Parameter	BATCHDAEMON
Description	Whether the Online Batch Deamon is enabled or not.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BATCHEDIT_ENABLED - Whether Batch Edit Facility is Enabled

Parameter	BATCHEDIT_ENABLED
Description	Whether the Batch Edit facility is enabled or not.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCHENABLED - Whether Online Batch Server is Enabled

Parameter	BATCHENABLED
Description	Whether the Online Batch Server is enabled or not.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BATCHTHREADS - Number of Online Batch Server Threads

Parameter	BATCHTHREADS
Description	Number of Threads in DEFAULT online threadpool.
Ordinality	Mandatory
Valid Values	Default: **5**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BATCH_DBPASS - Batch User Password

Parameter	BATCH_DBPASS
Description	Encrypted password for database user used for batch.
Ordinality	Mandatory
Valid Values	Encrpyted password
J2EE Web Application Server	Not Applicable
Tier	BATCH DB
Installation Mode	Not Applicable

BATCH_DBUSER - Batch User

Parameter	BATCH_DBUSER
Description	Database user used for batch.
Ordinality	Mandatory
Valid Values	Valid database user
J2EE Web Application Server	Not Applicable

Tier	BATCH DB
Installation Mode	Not Applicable

BATCH_MEMORY_ADDITIONAL_OPT - Threadpool Worker JVM additional options

Parameter	BATCH_MEMORY_ADDITIONAL_OPT
Description	Additional JVM Options for Threadpools
Ordinality	Mandatory
Valid Values	Valid Threadpool parameters. For example: `BATCH_MEMORY_ADDITIONAL_OPT=-Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,address=7757,server=y,suspend=n -Djava.compiler=NONE`
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCH_MEMORY_OPT_MAX - Threadpool Worker Java Maximum Heap Size

Parameter	BATCH_MEMORY_OPT_MAX
Description	Maximum JVM Memory for each Threadpool instance
Ordinality	Mandatory
Valid Values	Memory specification (in MB). Default: **1024**
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCH_MEMORY_OPT_MAXPERMSIZE - Threadpool Worker Java Maximum Perm Size

Parameter	BATCH_MEMORY_OPT_MAXPERMSIZE
Description	Maximum JVM PermGen for each Threadpool instance
Ordinality	Mandatory
Valid Values	Memory specification (in MB). Default: **192**
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCH_MEMORY_OPT_MIN - Threadpool Worker Java Minimum Heap Size

Parameter	BATCH_MEMORY_OPT_MIN
Description	Minimum JVM Memory for each Threadpool instance
Ordinality	Mandatory
Valid Values	Memory specification (in MB). Default: **512**
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCH_MODE - Default Mode of Batch

Parameter	BATCH_MODE
Description	Batch Execution Mode for Threadpools
Ordinality	Mandatory
Valid Values	[DISTRIBUTED|**CLUSTERED**]
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BATCH_RMI_PORT - RMI Port for Batch

Parameter	BATCH_RMI_PORT
Description	Port number used for communication to Threadpools. This is used for JMX monitoring.
Ordinality	Mandatory
Valid Values	Valid Port Number
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

BSN_APP - Business Server Application Name

Parameter	BSN_APP
Description	Deployed Business Application Server Name
Ordinality	Mandatory
Valid Values	Default: **SPLService**

J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_JMX_RMI_PORT_PERFORMANCE - JMX Port for Business Application Server

Parameter	BSN_JMX_RMI_PORT_PERFORMANCE
Description	JMX Port for Business Application Server monitoring
Ordinality	Optional
Valid Values	Valid open port for JMX
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_JMX_SYSPASS - Password for BAS JMX Monitoring

Parameter	BSN_JMX_SYSPASS
Description	Default encrypted password for JMX monitoring of Business Application Server Name for BSN_JMX_SYSUSER. This is required if BSN_JMX_RMI_PORT_PERFORMANCE is set.
Ordinality	Optional
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_JMX_SYSUSER - Default User for BSN JMX Monitoring

Parameter	BSN_JMX_SYSUSER
Description	Default user for JMX monitoring of Business Application Server Name. This is required if BSN_JMX_RMI_PORT_PERFORMANCE is set.
Ordinality	Optional
Valid Values	Administration User
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_JVMCOUNT - Number of Child JVM's

Parameter	BSN_JVMCOUNT
Description	Number of Child JVM's to maintain for COBOL to Java communication. This setting is only for products supporting COBOL based extensions.
Ordinality	Mandatory
Valid Values	Default:**2**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_NODENAME - IBM WebSphere Node Name

Parameter	BSN_NODENAME
Description	IBM WebSphere Node to deploy Business Application Server to.
Ordinality	Mandatory
Valid Values	Valid IBM WebSphere Node Names
J2EE Web Application Server	WAS WASND
Tier	BAS
Installation Mode	EMBED

BSN_RMIPORT - RMI Port for Child JVM

Parameter	BSN_RMIPORT
Description	Starting Port Number for Child JVM's to use for communication. This setting is only for products supporting COBOL based extensions.
Ordinality	Mandatory
Valid Values	Valid Starting port number. Ports starting from this number must be open on the machine housing the Child JVM's.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_SVRNAME - IBM WebSphere Server Name

Parameter	BSN_SVRNAME
Description	IBM WebSphere Server Name used to deploy Business Application Server to.
Ordinality	Mandatory
Valid Values	Valid IBM WebSphere Server Names
J2EE Web Application Server	WAS WASND
Tier	BAS
Installation Mode	EMBED

BSN_WASBOOTSTRAPPORT - Bootstrap Port

Parameter	BSN_WASBOOTSTARTPPORT
Description	IBM WebSphere Server Bootstrp Port used for Business Application Server.
Ordinality	Mandatory
Valid Values	Valid IBM WebSphere Bootstrap Ports
J2EE Web Application Server	WAS WASND
Tier	BAS
Installation Mode	EMBED

BSN_WLHOST - Business App Server Host

Parameter	BSN_WLHOST
Description	Oracle WebLogic host name for Business Application Server. For native and/or clustered installations this setting should be set to `localhost`.
Ordinality	Mandatory
Valid Values	Default: Current Host Name
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

BSN_WLS_SVRNAME - Oracle WebLogic Server Name

Parameter	BSN_WLS_SVRNAME

Description	Oracle WebLogic server name for Business Application Server. For native and/or clustered installations this setting should be set to the server or cluster created for the deployment.
Ordinality	Mandatory
Valid Values	Default: **myserver**
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

CHILD_JVM_JAVA_HOME - Child JVM Home Location

Parameter	CHILD_JVM_JAVA_HOME
Description	Location of Java Home used for Child JVM's. This is to support different JVM runtimes for Child JVM's. This setting is only for products supporting COBOL based extensions.
Ordinality	Mandatory
Valid Values	Valid JAVA_HOME.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

CHILD_JVM_PATH - Location of Child JVM Libraries

Parameter	CHILD_JVM_PATH
Description	Autogenerated location of Java libraries used for Child JVM's. As this setting is autogenerated it should not be manually altered. This setting is only for products supporting COBOL based extensions.
Ordinality	Mandatory
Valid Values	Generated Library Path.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

CLEANSE_INTERVAL - Mobile Registry Refresh Rate

Parameter	**CLEANSE_INTERVAL**
Description	Prefresh time, in seconds, for scheduler cache with product.
Ordinality	Mandatory
Valid Values	See spl.mwm.scheduler.cleanse.interval - Scheduler Cache Clense
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

CMPDB - Database Type

Parameter	**CMPDB**
Description	Database Type. This parameter is provided for backward compatibility and is set to **ORACLE**. This should not be altered.
Ordinality	Mandatory
Valid Values	**ORACLE**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH MOB
Installation Mode	EMBED NATIVE

COBDIR - COBOL Home Directory

Parameter	**COBDIR**
Description	Location of the COBOL runtime. This parameter is only application for products that support COBOL based extensions.
Ordinality	Mandatory
Valid Values	Location of COBOL installation
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH
Installation Mode	EMBED NATIVE

COHERENCE_CLUSTER_ADDRESS - Multicast address for Batch Cluster

Parameter	**COHERENCE_CLUSTER_ADDRESS**

Description	Multicast IP Address for **CLUSTERED** mode. Specify the multicast IP address that a Socket will listen or publish on.
Ordinality	Mandatory
Valid Values	Valid values are from 224.0.0.0 to 239.255.255.255.
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

COHERENCE_CLUSTER_MODE - Batch Cluster Mode

Parameter	**COHERENCE_CLUSTER_MODE**
Description	Oracle Coherence mode. Coherence may be configured to operate in either development or production mode. These modes do not limit access to features, but instead alter some default configuration settings. For instance, development mode allows for faster cluster startup to ease the development process. The development mode is used for all pre-production activities, such as development and testing. This is an important safety feature because development nodes are restricted from joining with production nodes. Development mode is the default mode. Production mode (**prod**) must be explicitly specified when using Coherence in a production environment
Ordinality	Mandatory
Valid Values	[**dev**|prod]
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

COHERENCE_CLUSTER_NAME - Batch Cluster Name

Parameter	**COHERENCE_CLUSTER_NAME**
Description	Unique Cache name for Coherence Clusters.
Ordinality	Mandatory
Valid Values	Any string value. It is recommended to use the **SPLENVIRON** value, optionally, with the database schema owner as a unique identifier. For example: DEV.SPLADM

J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

COHERENCE_CLUSTER_PORT - Batch Cluster Port Number

Parameter	COHERENCE_CLUSTER_PORT
Description	Unique Cache port for the Coherence Cluster.
Ordinality	Mandatory
Valid Values	Any open relevant port unique to the COHERENCE_CLUSTER_NAME.
J2EE Web Application Server	Not Applicable
Tier	BATCH
Installation Mode	Not Applicable

CONTEXTFACTORY - Context Factory for Mobile Application

Parameter	CONTEXTFACTORY
Description	ContextFactory used for Mobile Integration
Ordinality	Mandatory
Valid Values	Refer to spl.mwm.abr.contextFactory - ABR Context Factory for more details.
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

DB_OVERRIDE_CONNECTION - Custom JDBC URL

Parameter	DB_OVERRIDE_CONNECTION
Description	Database JDBC Override string. For customers using Oracle 12c, use a string in the following format to support Pluggable Databases: `jdbc:oracle:thin:@//<host>:<port>/<pdbname>` Customers using Native mode installation can replace this connection with JDBC Data Sources, if desired.
Ordinality	Mandatory
Valid Values	Valid JDBC URL.

J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBCONNECTION - JDBC Connection string

Parameter	DBCONNECTION
Description	Generated Database JDBC string. This is generated from the DBSERVER / DBPORT / DBNAME or DB_OVERRIDE_CONNECTION. As this is generated, it should not be altered. Customers using Native mode installation can replace this connection with JDBC Data Sources, if desired.
Ordinality	Mandatory
Valid Values	Generated URL
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBDRIVER - Database Driver

Parameter	DBDRIVER
Description	Database Driver to use. Automatically generated by installation. This should not be altered.
Ordinality	Mandatory
Valid Values	Generated URL
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPASS - Online Database User Password

Parameter	DBPASS
Description	Encrypted database user password for the online component of the product. Password for DBUSER.
Ordinality	Mandatory

Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPASS_GEOCODE_WLS - GEOCODE Database Password

Parameter	DBPASS_GEOCODE_WLS
Description	Encrypted database user password for the GEOCODE integration. Password for DBUSER_GEOCODE.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	MOB DB
Installation Mode	EMBED NATIVE

DBPASS_MDS - MDS Database Password

Parameter	DBPASS_MDS
Description	Encrypted database user password for the SOA Suite Meta Data Service (MDS) integration. Password for DBUSER_MDS. This password was specified as part of the SOA Suite database installation process for the *prefix*_MDS user. This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPASS_ORASDPM - DPM Database Password

Parameter	DBPASS_ORASDPM
Description	Encrypted database user password for the SOA Suite User Message Services (DPM) integration. Password for

	DBUSER_ORASDPM. This password was specified as part of the SOA Suite database installation process for the ***prefix*_ORASDPM** user. This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPASS_OSB - OSB Database Password

Parameter	**DBPASS_OSB**
Description	Encrypted database user password for the Oracle Service Bus (OSB) integration. Password for **DBUSER_OSB**. This password was specified as part of the Oracle Service Bus installation for the OSB database user. This is only valid for products that include Oracle Service Bus. This is used for deployment of content to Oracle Service Bus for integration. This is not used for the Oracle Service Bus Adapters for Oracle Utilities.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPASS_SOAINFRA - SOA Database Password

Parameter	**DBPASS_SOAINFRA**
Description	Encrypted database user password for the SOA Suite integration. Password for **DBUSER_SOAINFRA**. This password was specified as part of the SOA Suite database installation process for the ***prefix*_SOAINFRA** user.

	This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBPORT - Database Listener Port

Parameter	DBPORT
Description	Oracle Listener port for connection to the database. Customers using Native mode installation can ignore this value if using JDBC Data Sources.
Ordinality	Mandatory
Valid Values	Valid Port Number
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBSERVER - Database Server

Parameter	DBSERVER
Description	Oracle Database Host Name. Customers using Native mode installation can ignore this value if using JDBC Data Sources.
Ordinality	Mandatory
Valid Values	Valid Host Name
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBURL_GEOCODE - GEOCODE JDBC URL

Parameter	DBURL_GEOCODE
Description	Oracle Database JDBC URL used for Geocoding.

	This is set for products using Oracle Spatial.
Ordinality	Mandatory
Valid Values	Valid JDBC URL
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBURL_OSB - OSB JDBC URL

Parameter	DBURL_OSB
Description	Oracle Database JDBC URL for Oracle Service Bus. This is used for deploying content to Oracle Service Bus.
Ordinality	Mandatory
Valid Values	Valid JDBC URL
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBURL_SOA - SOA Suite JDBC URL

Parameter	DBURL_SOA
Description	Oracle Database JDBC URL for Oracle SOA Suite. This is used for deploying content to Oracle SOA Suite.
Ordinality	Mandatory
Valid Values	Valid JDBC URL
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER - Online Database User

Parameter	DBUSER
Description	Database User, with Read/Write permissions used for pooled connections for online transactions.
Ordinality	Mandatory
Valid Values	Valid Database User used in installation

J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER_GEOCODE - GEOCODE Database User

Parameter	DBUSER_GEOCODE
Description	Database User, with connection to Geocode Database.
Ordinality	Mandatory
Valid Values	Valid Database User on target database
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER_MDS - SOA MDS Database User

Parameter	DBPASS_MDS
Description	Database user password for the SOA Suite Meta Data Services (MDS) integration. This userid was specified as part of the SOA Suite database installation process for the *prefix*_MDS user. This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Valid Database User (*prefix*_MDS) on target database
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER_ORASDPM - DPM Database User

Parameter	DBPASS_ORASDPM
Description	Database user password for the SOA Suite User Messaging Services (DPM) integration. This userid was specified as part of the SOA Suite database installation process for the *prefix*_ORASDPM user. This is only valid for products that include Oracle SOA

	Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Valid Database User (***prefix*_ORASDPM**) on target database
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER_OSB - OSB Database User

Parameter	**DBPASS_OSB**
Description	Database user password for the Oracle Service Bus integration. This userid was specified as part of the Oracle Service Bus database installation process. This is only valid for products that include Oracle Service Bus. This is used for deployment of content to Oracle Service Bus for integration.
Ordinality	Mandatory
Valid Values	Valid Database User on target database
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DBUSER_SOAINFRA - SOA Database User

Parameter	**DBPASS_SOAINFRA**
Description	Database user password for the SOA Suite integration. This userid was specified as part of the SOA Suite database installation process for the ***prefix*_SOAINFRA** user. This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.
Ordinality	Mandatory
Valid Values	Valid Database User (***prefix*_SOAINFRA**) on target database
J2EE Web Application Server	WLS WAS WASND

Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DESC - Environment Description

Parameter	DESC
Description	Environment Description. Used for documentation purposes only.
Ordinality	Optional
Valid Values	String Value
J2EE Web Application Server	Not Applicable
Tier	Not Applicable
Installation Mode	Not Applicable

DIALECT - Hibernate Dialect

Parameter	DIALECT
Description	Hibernate Dialect This is the database dialect used by Hibernate to process database requests. This is automatically generated by the installer and should not be altered.
Ordinality	Mandatory
Valid Values	Valid Hibernate Dialects
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

DIRSEP - Directory separator

Parameter	DIRSEP
Description	Directory Seperator This is an internal setting that sets the directory separator applicable for the platform. This setting is used by various scripts to set the paths correctly for the operating system. This setting is automatically generated at installation time and should not be altered.
Ordinality	Mandatory
Valid Values	Directory Seperator character

J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

DOC1BILLSCRIPT - Name Of External Print Script

Parameter	DOC1BILLSCRIPT
Description	This setting sets the name of the external printing script used for printing extracts using Oracle Documaker. This is generated by individual products that support this feature.
Ordinality	Mandatory
Valid Values	Name of supplied script
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

DOC1SCRIPT - Name Of Generic Print Script

Parameter	DOC1SCRIPT
Description	This setting sets the name of the generic printing script used for extracts using Oracle Documaker. This is generated by individual products that support this feature.
Ordinality	Mandatory
Valid Values	Name of supplied script
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

ENCODING - Whether Encryption Is Enabled

Parameter	ENCODING
Description	Enables or Disables encryption across the product. By default encryption is enabled (**true**) and is recommended to be enabled for all environments.
Ordinality	Mandatory
Valid Values	[**true** | false]

J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

FW_VERSION - Oracle Utilities Application Framework version

Parameter	FWVERSION
Description	Oracle Utilities Application Version This is an internal setting that is generated upon installation of the product, service pack installation or upgrade installation. This setting is automatically generated at installation time and should not be altered.
Ordinality	Mandatory
Valid Values	Valid Oracle Utilities Application Framework version
J2EE Web Application Server	Not Applicable
Tier	Not Applicable
Installation Mode	Not Applicable

FW_VERSION_NUM - Oracle Utilities Application Framework Version Identifier

Parameter	FW_VERSION_NUM
Description	Oracle Utilities Application Version This is an internal setting that is generated upon installation of the product, service pack installation or upgrade installation. This setting is automatically generated at installation time and should not be altered.
Ordinality	Mandatory
Valid Values	Valid Oracle Utilities Application Framework version
J2EE Web Application Server	Not Applicable
Tier	Not Applicable
Installation Mode	Not Applicable

GIS - GIS Support

Parameter	GIS
Description	Enables or Disables GIS support within the product.

	This setting is only used by products that use GIS integration. Setting this value to **true** enabled all other GIS settings.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS
Tier	WEB BAS BATCH
Installation Mode	EMBED NATIVE

GIS_URL - GIS Service URL

Parameter	**GIS_URL**
Description	Specifies the GIS JNDI URL used for connections. This setting is only used by products that use GIS integration. This setting is only used if **GIS** is set to true.
Ordinality	Mandatory
Valid Values	Valid JNDI URL
J2EE Web Application Server	WLS
Tier	WEB BAS BATCH
Installation Mode	EMBED NATIVE

GIS_WLSYSPASS - GIS WebLogic System Password

Parameter	**GIS_WLSYSPASS**
Description	Encrypted JNDI Password for GIS integration. This setting is only used by products that use GIS integration. This setting is only used if **GIS** is set to true.
Ordinality	Mandatory
Valid Values	Valid Encrypted Password
J2EE Web Application Server	WLS
Tier	WEB BAS BATCH
Installation Mode	EMBED NATIVE

GIS_WLSYSUSER - GIS WebLogic System User Id

Parameter	GIS_WLSYSUSER
Description	Encrypted JNDI User for GIS integration. This setting is only used by products that use GIS integration. This setting is only used if GIS is set to true.
Ordinality	Mandatory
Valid Values	Valid Encrypted User
J2EE Web Application Server	WLS
Tier	WEB BAS BATCH
Installation Mode	EMBED NATIVE

help - Name of online help WAR file

Parameter	help
Description	Name of Help WAR file. Used in build of help application.
Ordinality	Mandatory
Valid Values	This value is automatically generated
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

HEADEND_CD_CB - Headend System URI for SOA Configuration Plan (CD_CB)

Parameter	HEADEND_CD_CB
Description	URI as configure in SOA Suite Configuration Plan for Headend System for the CD_DB interface (Echelon). This setting is only available to products interfacing to head end systems using SOA Suite.
Ordinality	Mandatory
Valid Values	Valid SOA Suite URI for CD_DB interface.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

HEADEND_MR_CB - Headend System URI for SOA Configuration Plan (MR_CB)

Parameter	HEADEND_MR_CB
Description	URI as configure in SOA Suite Configuration Plan for Headend System for the MR_DB interface (Echelon). This setting is only available to products interfacing to head end systems using SOA Suite.
Ordinality	Mandatory
Valid Values	Valid SOA Suite URI for MR_DB interface.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

HIBERNATE_JAR_DIR - Location of Hibernate JAR files

Parameter	HIBERNATE_JAR_DIR
Description	Location of the Hibernate JAR files. This setting is used by the installer to copy the relevant files to the desired locations for deployment. This setting is only used at installation time but is registered for potential post release use.
Ordinality	Mandatory
Valid Values	Valid location of Hibernate JAR Files as outlined in the product Installation Guide.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH IWS
Installation Mode	EMBED NATIVE

HIGHVALUE - Language specific highvalues

Parameter	HIGHVALUE
Description	High value used for SQL queries. This is autogenerated at installation time.
Ordinality	Mandatory
Valid Values	This value is automatically generated and should not be altered.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB

Installation Mode	EMBED NATIVE

IPCSTARTPORT - Starting IPC Port for Scheduler

Parameter	IPCSTARTPORT
Description	Starting port number for Interprocess Communcications (IPC) for Scheduler
Ordinality	Mandatory
Valid Values	Valid open port numbers
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

JAVA_HOME - Location of Java SDK

Parameter	JAVA_HOME
Description	Location of JDK for use with product
Ordinality	Mandatory
Valid Values	Valid JAVA_HOME location
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

JAVAENCODING - Java Language Encoding

Parameter	JAVAENCODING
Description	Encoding string for Java to support appropriate character sets.
Ordinality	Mandatory
Valid Values	Valid Java Encoding (java.lang). Default: **UTF8**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

JNDI_GEOCODE - Geocode Data Source JNDI

Parameter	JNDI_GEOCODE
Description	JNDI name for the Geocode Data Source. This parameter is

	used with for the Geocode interface. This parameter is only valid for products that use the Geocode interface.
Ordinality	Mandatory
Valid Values	Valid JNDI for Geocode Data Source
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH MOB
Installation Mode	EMBED NATIVE

JNDI_OSB - OSB Data Source JNDI

Parameter	JNDI_OSB
Description	JNDI name for the Oracle Service Bus Data Source. This parameter is used with for the Oracle Service Bus interface. This parameter is only valid for products that ship integration that requires Oracle Service Bus.
Ordinality	Mandatory
Valid Values	Valid JNDI for Oracle Service Bus Data Source
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH
Installation Mode	EMBED NATIVE

JVM_ADDITIONAL_OPT - Child JVM additional Options

Parameter	JVM_ADDITIONAL_OPT
Description	Additional Java options for the Child JVM. Allows implementations to add additional java options for the Child JVM. This parameter is only valid for product that support COBOL based extensions.
Ordinality	Mandatory
Valid Values	Valid Java options (refer JVM vendor documentation)
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

JVMCOMMAND - Generated Child JVM Command

Parameter	JVMCOMMAND
Description	Command executed to initiate to start Child JVM. This paremeter is automatically generated. This parameter is only valid for product that support COBOL based extensions.
Ordinality	Mandatory
Valid Values	Generated command line
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

JVMMEMORYARG - Child JVM Memory Allocation

Parameter	JVMCOMMAND
Description	Maximum memory (in MB) to be allocated per Child JVM. This parameter is only valid for product that support COBOL based extensions.
Ordinality	Mandatory
Valid Values	Valid memory specification. Default: **512**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

KS_ALIAS - Key Alias

Parameter	KS_ALIAS
Description	Key Alias contained in keystore used by keytool. It is recommended to retain the default setting, unless an alternative is desired.
Ordinality	Mandatory
Valid Values	Valid alias. Default: **ouaf.system**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_ALIAS_KEYALG - Key Algorithm

Parameter	KS_ALIAS_KEYALG
Description	Key Alias Algorithm used by keystore used by keytool.
Ordinality	Mandatory
Valid Values	Valid algorithm. Default: **AES**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_ALIAS_KEYSIZE - Keysize

Parameter	KS_ALIAS_KEYSIZE
Description	Keysize strength used by keystore used by keytool.
Ordinality	Mandatory
Valid Values	Valid keysize. Default: **128**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_HMAC_ALIAS - HMAC Key Alias

Parameter	KS_HMAC_ALIAS
Description	Key HMAC Alias contained in keystore used by keytool. It is recommended to retain the default setting, unless an alternative is desired.
Ordinality	Mandatory
Valid Values	Valid alias. Default: **ouaf.system.hmac**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_HMAC_ALIAS_KEYALG - HMAC Key Algorithm

Parameter	KS_HMAC_ALIAS_KEYALG
Description	Key HMAC Alias Algorithm used by keystore used by keytool.

Ordinality	Mandatory
Valid Values	Valid algorithm. Default: **HmacSHA256**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_HMAC_ALIAS_KEYSIZE - HMAC Keysize

Parameter	KS_HMAC_ALIAS_KEYSIZE
Description	HMAC Keysize strength used by keystore used by keytool.
Ordinality	Mandatory
Valid Values	Valid keysize. Default: **256**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_KEYSTORE_FILE - Keystore File Name

Parameter	KS_KEYSTORE_FILE
Description	Keystore File used by keytool.
Ordinality	Mandatory
Valid Values	Valid location and file name for keystore. Linux: $SPLEBASE/ks/.ouaf_keystore Windows: %SPLEBASE%\ks\.ouaf_keystore
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_MODE - Key Padding Mode

Parameter	KS_MODE
Description	Key Padding Mode
Ordinality	Mandatory
Valid Values	Valid Padding for Keystore. Default: CBC

J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_PADDING - Key Padding Algorithm

Parameter	KS_PADDING
Description	Key Padding Algorithm used for key generation
Ordinality	Mandatory
Valid Values	Valid Padding Algorithm for Keystore. Default: **PKCS5Padding**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_STOREPASS_FILE - Key Password Store Pass

Parameter	KS_STOREPASS_FILE
Description	Password Keystore File used by keytool.
Ordinality	Mandatory
Valid Values	Valid location and file name for keystore. Linux: $SPLEBASE/ks/.ouaf_storepass Windows: %SPLEBASE%\ks\.ouaf_storepass
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

KS_STORETYPE - Key Store Type

Parameter	KS_STORETYPE
Description	Keystore Store Type used by keytool.
Ordinality	Mandatory
Valid Values	Valid Key Store Type. Default: **JCEKS**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB

Installation Mode	EMBED NATIVE

LD_LIBRARY_PATH - Library Path for Windows/Linux/Solaris

Parameter	LD_LIBRARY_PATH
Description	Generated Library Path for Windows, Linux and Solaris
Ordinality	Mandatory
Valid Values	This value is generated by the product installer.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

LIBPATH - Library Path for AIX

Parameter	LIBPATH
Description	Generated Library Path for AIX
Ordinality	Mandatory
Valid Values	This value is generated by the product installer.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

MAPDIR - Location of Map files used for Scheduler

Parameter	MAPDIR
Description	Location of local Map file used by Scheduler. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	Fuly qualified directory holding map files
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MAPVIEWER_EAR - Location of Mapviewer EAR file

Parameter	MAPVIEWER_EAR

Description	Location of MapViewer EAR file. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	Fully qualified directory EAR file
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MAPVIEWER_ISLOCAL - Deploy Mapviewer Locally

Parameter	MAPVIEWER_ISLOCAL
Description	Whether the MapViewer application is to be deployed within the same server as the product. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	[true | **false**]
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MAXPROCESSINGTIME - Maximum Scheduler Processing Time

Parameter	MAXPROCESSINGTIME
Description	Transaction timeout, in seconds, for each appointment booking processed by Scheduler. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **5**)
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MINREQUESTS - Initial Scheduler Minimum Requests

Parameter	MINREQUESTS

Description	Initial request pool size for Scheduler. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **1**)
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MOBILITY_APP_ONLY - Deploy Only Mobility Web Application

Parameter	MOBILITY_APP_ONLY
Description	Whether the mobile web application is the only deployment on the server. This allows for a distributed installation. This parameter only applies to products that use the Scheduler.
Ordinality	Mandatory
Valid Values	[true | **false**]
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

MODULES - Names of Modules installed

Parameter	MODULES
Description	Generated list of modules. This is generated by the product installer. This parameter is provided for backward compatibility as modules installed are now autogenerated from installation records.
Ordinality	Optional
Valid Values	List of product codes.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS BATCH MOB
Installation Mode	EMBED NATIVE

NLS_LANG - NLS Language setting

Parameter	`NLS_LANG`
Description	Globalization Language setting for the database.
Ordinality	Mandatory
Valid Values	Valid NLS_LANG. Default: `AMERICA_AMERICAN.AL32UTF8`
J2EE Web Application Server	WLS WAS WASND
Tier	BAS IWS BATCH DB
Installation Mode	EMBED NATIVE

NODEID - Scheduler Node Identifier

Parameter	`NODEID`
Description	Scheduler Node Identifier. This parameter is provided for backward compatibility. This parameter only applies to products using the Scheduler.
Ordinality	Optional
Valid Values	Valid Node Identifier
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

OIM_SPML_NAME_SPACE - OIM Default Namespace

Parameter	`OIM_SPML_NAME_SPACE`
Description	Default Namespace used for Oracle Identity Manager integration. This setting is only available if `OPEN_SPML_ENABLED_ENV` is `true`.
Ordinality	Optional
Valid Values	Valid SPML Namespace. Default: **http://xmlns.oracle.com/OIM/provisioning**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

OIM_SPML_SOAP_DEBUG_SETTING - OIM Interface Debug

Parameter	OIM_SPML_SOAP_DEBUG_SETTING
Description	Enables or Disables debug mode to diagnose issues with the Oracle Identity Manager integration. This setting is only available if OPEN_SPML_ENABLED_ENV is true.
Ordinality	Optional
Valid Values	[true | **false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

OIM_SPML_SOAP_ELEMENT - OIM SOAP Element

Parameter	OIM_SPML_SOAP_ELEMENT
Description	Name of top level SOAP element used for Oracle Identity Manager interface. This is autogenerated. This setting is only available if OPEN_SPML_ENABLED_ENV is true.
Ordinality	Optional
Valid Values	Valid SOAP Element. Default: **sOAPElement**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

OIM_SPML_UBER_SCHEMA_NAME - Name of OIM Interface Schema

Parameter	OIM_SPML_UBER_SCHEMA_NAME
Description	Name of schema used for Oracle Identity Manager interface. This is autogenerated. This setting is only available if OPEN_SPML_ENABLED_ENV is true.
Ordinality	Optional
Valid Values	Valid product Schema. Default: **F1-IDMUser**
J2EE Web Application Server	WLS WAS WASND
Tier	BAS

Installation Mode	EMBED NATIVE

ONLINE_DISPLAY_HOME - Location of Print Rendering Software

Parameter	ONLINE_DISPLAY_HOME
Description	Location of Print Rendering Software such as Oracle Documaker or Group 1 Doc 1.
Ordinality	Optional
Valid Values	Valid product location of the print rendering software
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH
Installation Mode	EMBED NATIVE

ONS_JAR_DIR - Location of Oracle Notification Service Libraries

Parameter	ONS_JAR_DIR
Description	Location of Oracle Notification Services Libraries used for RAC Fast Connection Failover support.
Ordinality	Optional
Valid Values	Valid location of ons.jar file. Usually `$ORACLE_HOME/opmn/lib`
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

ONSCONFIG - ONS Configuration

Parameter	ONSCONFIG
Description	ONS configuration string used for RAC Fast Connection Failover support. For example: `racnode1.my.com:4200,racnode2.my.com:4200`
Ordinality	Optional
Valid Values	ONS configuration string without nodes= prefix.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH DB
Installation Mode	EMBED NATIVE

OPEN_SPML_ENABLED_ENV - Whether OIM Interface enabled

Parameter	OPEN_SPML_ENABLED_ENV
Description	Enable or Disable Oracle Identity Manager integration
Ordinality	Optional
Valid Values	[true | **false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

OPSYS - Operating System

Parameter	OPSYS
Description	Internal variable used for platform specific configurations to be implemented.
Ordinality	Mandatory
Valid Values	Generated Operating System String
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

ORACLE_CLIENT_HOME - Oracle Client Home

Parameter	ORACLE_CLIENT_HOME
Description	Location of Oracle Client Software or Oracle Home. This is used for ONS and/or the Perl runtime.
Ordinality	Mandatory
Valid Values	Location of Oracle Client Installation. If the Oracle Database software is installed on the same machine, then this can be ORACLE_HOME.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

OSB_HOME - Oracle Service Bus Home

Parameter	OSB_HOME
Description	Location of Oracle Service Bus (WL_HOME). This is used to

	deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Location of Oracle Service Bus installation.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

OSB_HOST - Oracle Service Bus Host Name

Parameter	OSB_HOST
Description	Host Name for Oracle Service Bus. This is used to deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Host Name for Oracle Service Bus.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

OSB_LOG_DIR - Default Location For Oracle Service Bus Logs

Parameter	OSB_LOG_DIR
Description	Loggoing Directory Oracle Service Bus. This is used to deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Logging directory for Oracle Service Bus content deployment.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

OSB_PASS_WLS - Password for Oracle Service Bus User

Parameter	**OSB_PASS_WLS**
Description	Encrypted password for OSB deployment user (**OSB_USER**). This is used to deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Generated Password.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

OSB_PORT_NUMBER - Oracle Service Bus Port Number

Parameter	**OSB_PORT_NUMBER**
Description	Port number allocated to Oracle Service Bus for . This is used to deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Valid Administration port to use for deployment of content.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

OSB_USER - User Name for Oracle Service Bus

Parameter	**OSB_USER**
Description	Administration user used for OSB deployment user. This is used to deploy Oracle Service Bus content. This setting only applies to products with prebuilt Oracle Service Bus content. This is not used for Oracle Service Bus Adapters.
Ordinality	Optional
Valid Values	Valid Administration user to use for deployment of content.

J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

PERLHOME - Location of PERL

Parameter	PERLHOME
Description	Generated location for Perl runtime. Generated from ORACLE_CLIENT_HOME.
Ordinality	Mandatory
Valid Values	Generated location for Perl.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

REL_CBL_THREAD_MEM - Release COBOL Memory

Parameter	REL_CBL_THREAD_MEM
Description	Generated string to enable reset of COBOL memory. This setting is only applicable for products supporting COBOL based objects.
Ordinality	Mandatory
Valid Values	Generated options.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

RJVM - Enable Child JVM

Parameter	RJVM
Description	Enable or Disable Child JVM. This setting is only applicable for products supporting COBOL based objects. This must be set to **true** for COBOL support.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

SERVER_EXPRESS_COBDIR - Location of Server Express

Parameter	SERVER_EXPRESS_COBDIR
Description	Location of COBOL Compilter. This setting is only applicable for products supporting COBOL based objects and require Server Express to compile custom COBOL objects.
Ordinality	Optional
Valid Values	Valid location of Microfocus Server Express.
J2EE Web Application Server	WLS WAS WASND
Tier	BAS BATCH
Installation Mode	EMBED NATIVE

SOA_HOME - Location Of Oracle SOA Suite

Parameter	SOA_HOME
Description	Location of Oracle SOA Suite software. This is used to deploy Oracle SOA Suite content. This setting only applies to products with prebuilt Oracle SOA Suite content.
Ordinality	Optional
Valid Values	Valid location of Oracle SOA Suite.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

SOA_HOST - Hostname for Oracle SOA Suite

Parameter	SOA_HOST
Description	Hostname for Oracle SOA Suite software. This is used to deploy Oracle SOA Suite content. This setting only applies to products with prebuilt Oracle SOA Suite content.
Ordinality	Optional
Valid Values	Valid host name for Oracle SOA Suite.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

SOA_PORT_NUMBER - Port Number for Oracle SOA Suite

Parameter	SOA_PORT_NUMBER
Description	Port Number for Oracle SOA Suite software. This is used to deploy Oracle SOA Suite content. This setting only applies to products with prebuilt Oracle SOA Suite content.
Ordinality	Optional
Valid Values	Valid port number for Oracle SOA Suite.
J2EE Web Application Server	WLS
Tier	BAS
Installation Mode	EMBED NATIVE

SPLADMIN - Administration Userid

Parameter	SPLADMIN
Description	Administration OS user used to install the software. Used by utilities to assign ownership and for deployment purposes.
Ordinality	Mandatory
Valid Values	Valid Administration Account.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

SPLADMINGROUP - Administration Group

Parameter	SPLADMINGROUP
Description	Administration OS group used to install the software. Used by utilities to assign ownership and for deployment purposes.
Ordinality	Mandatory
Valid Values	Valid Administration Group.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

SPLApp - Name of Online WAR file

Parameter	SPLApp
Description	Name of online WAR file used for building purposes
Ordinality	Mandatory
Valid Values	Valid WAR file name.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS XAI
Installation Mode	EMBED NATIVE

SPLDIR - Home Directory Of Product

Parameter	SPLDIR
Description	Home Location of product software. This value must match the corresponding entry in cistab.
Ordinality	Mandatory
Valid Values	Valid Location of software. Generated by installer.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

SPLDIROUT - Location Of Product Output

Parameter	SPLDIROUT
Description	Home Location of product output. This value must match the corresponding entry in cistab.
Ordinality	Mandatory
Valid Values	Valid Location of output. Generated by installer.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

SPLENVIRON - Environment Identifier

Parameter	SPLDIROUT
Description	Home Location of product output. This value must match the corresponding entry in cistab.
Ordinality	Mandatory
Valid Values	Valid Location of output. Generated by installer.

J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

SPLSERVICEAPP - Name of Business Application Server Application

Parameter	SPLSERVICEAPP
Description	Default Name for Business Application Application Name.
Ordinality	Mandatory
Valid Values	Valid Server Application Names
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

SPLWAS - J2EE Web Application Server Type

Parameter	SPLWAS
Description	J2EE Web Application Server to be used for product.
Ordinality	Mandatory
Valid Values	Following Valid Values: **WLS** - Oracle WebLogic **WAS** - IBM WebSphere **WASND** - IBM WebSphere ND
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED NATIVE

SPLWEBAPP - Name of Web Application Server Application

Parameter	SPLWEBAPP
Description	Default Name for Web Application Application Name.
Ordinality	Mandatory
Valid Values	Valid Server Application Names
J2EE Web Application Server	WLS WAS WASND
Tier	WEB XAI MOB
Installation Mode	EMBED NATIVE

STRIP_HTML_COMMENTS - Strip Out Comments In Code

Parameter	STRIP_HTML_COMMENTS
Description	Enable or Disable Stripping comments in generated code. By default, code is commented for ease of extension. If these comments represent a security issue for your site, then they can be stripped out at runtime (**true**).
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS WAS WASND
Tier	BAS
Installation Mode	EMBED NATIVE

TIMEOUT - Mobile Connection Timeout

Parameter	TIMEOUT
Description	Connection timeout, in seconds, for Mobile component between handhelds and server.
Ordinality	Mandatory
Valid Values	Valid Timeout value. Refer to spl.mwm.abr.timeout for more details.
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

TOP_VERSION - Product Version

Parameter	TOP_VERSION
Description	Product Version identifier string used by utilities and upgrade scripts.
Ordinality	Mandatory
Valid Values	Generated by installer and should not be altered manually.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

TOP_VERSION_NUM - High level Product Version

Parameter	TOP_VERSION_NUM
Description	Product Version identifier number used by utilities and upgrade scripts.
Ordinality	Mandatory
Valid Values	Generated by installer and should not be altered manually.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

URL - Mobile JNDI URL

Parameter	URL
Description	Connection URL, for Mobile component.
Ordinality	Mandatory
Valid Values	Valid URL. Refer to spl.mwm.abr.url for more details.
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

WAS_HOME - IBM WebSphere Home

Parameter	WAS_HOME
Description	Location of IBM WebSphere software for deployment purposes. This setting is only applicable to IBM WebSphere customers.
Ordinality	Mandatory
Valid Values	Location of IBM WebSphere installation
J2EE Web Application Server	WAS
Tier	WEB BAS IWS XAI MOB
Installation Mode	NATIVE

WAS_PASSWORD - IBM WebSphere Administration Password

Parameter	WAS_PASSWORD
Description	Encrpyted password for administration user for IBM WebSphere/IBM WebSphere ND. This setting is only

	applicable to IBM WebSphere and IBM WebSphere ND customers.
Ordinality	Mandatory
Valid Values	Encrypted password for administration user.
J2EE Web Application Server	WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	NATIVE

WAS_USERID - IBM WebSphere Administration User

Parameter	WAS_USERID
Description	Administration user for IBM WebSphere/IBM WebSphere ND. This setting is only applicable to IBM WebSphere and IBM WebSphere ND customers.
Ordinality	Mandatory
Valid Values	Administration user.
J2EE Web Application Server	WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	NATIVE

WASND_DMGR_HOST - IBM WebSphere Deployment Manager Host Name

Parameter	WASND_DMGR_HOST
Description	Location of IBM WebSphere ND host for deployment purposes. This setting is only applicable to IBM WebSphere ND customers.
Ordinality	Mandatory
Valid Values	Host of IBM WebSphere ND Deployment Manager
J2EE Web Application Server	WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	NATIVE

WASND_HOME - IBM WebSphere ND Home

Parameter	WASND_HOME
Description	Location of IBM WebSphere ND software for deployment purposes. This setting is only applicable to IBM WebSphere ND customers.

Ordinality	Mandatory
Valid Values	Location of IBM WebSphere ND installation
J2EE Web Application Server	WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	NATIVE

WEB_ADDITIONAL_OPT - Additional Java Options for Web

Parameter	WEB_ADDITIONAL_OPT
Description	Additional Java Options for Web Application Server. For native installations, these options can be set on the Server definition.
Ordinality	Optional
Valid Values	Valid java options for JVM used.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE - AppViewer Form Login Error Page

Parameter	WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE
Description	Error Page used for invalid AppViewer Login.
Ordinality	Mandatory
Valid Values	Valid JSP for error pages
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_APPVIEWER_FORM_LOGIN_PAGE - AppViewer Form Login Page

Parameter	WEB_APPVIEWER_FORM_LOGIN_PAGE
Description	Login Page used for AppViewer Login.
Ordinality	Mandatory
Valid Values	Valid JSP for login
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_APPVIEWER_PRINCIPAL_NAME - Application Viewer Principal Name

Parameter	WEB_APPVIEWER_PRINCIPAL_NAME
Description	Principal Name for AppViewer
Ordinality	Mandatory
Valid Values	Valid Principal Name
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_APPVIEWER_ROLE_NAME - Application Viewer Security Role

Parameter	WEB_APPVIEWER_ROLE_NAME
Description	Role Name for AppViewer
Ordinality	Mandatory
Valid Values	Valid Role Name
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_BATCH_CLUSTER_URL - Batch Cluster URL

Parameter	WEB_BATCH_CLUSTER_URL
Description	Batch Cluster URL for online submission tracking of Mobile application
Ordinality	Mandatory
Valid Values	Valid Role Name
J2EE Web Application Server	WLS WAS WASND
Tier	MOB
Installation Mode	EMBED NATIVE

WEB_CONTEXT_ROOT - Web Context Root

Parameter	WEB_CONTEXT_ROOT
Description	Web Context for product
Ordinality	Mandatory

Valid Values	Valid context. Default: **spl**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED NATIVE

WEB_FORM_LOGIN_ERROR_PAGE - Default Form Login Error Page

Parameter	WEB_FORM_LOGIN_ERROR_PAGE
Description	Error Page used for invalid online Login.
Ordinality	Mandatory
Valid Values	Valid JSP for error pages
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_FORM_LOGIN_PAGE - Default Form Login Page

Parameter	WEB_FORM_LOGIN_PAGE
Description	Login Page used for online Login.
Ordinality	Mandatory
Valid Values	Valid JSP for login
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_HELP_FORM_LOGIN_ERROR_PAGE - Help Form Login Error Page

Parameter	WEB_HELP_FORM_LOGIN_ERROR_PAGE
Description	Error Page used for invalid Help Login.
Ordinality	Mandatory
Valid Values	Valid JSP for error pages
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_HELP_FORM_LOGIN_PAGE - Help Form Login Page

Parameter	WEB_HELP_FORM_LOGIN_PAGE
Description	Login Page used for Help Login.
Ordinality	Mandatory
Valid Values	Valid JSP for login
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_ISAPPVIEWER - Enable AppViewer

Parameter	WEB_ISAPPVIEWER
Description	Enable or Disable AppViewer from deployment. The use of AppViewer is recommended for appropriate non-production environments only.
Ordinality	Mandatory
Valid Values	[**true** | false]
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_ISDEVELOPMENT - Development Environment

Parameter	WEB_ISDEVELOPMENT
Description	Enable or Disable Development on this product environment. This setting assumes that the Oracle Utilities SDK.
Ordinality	Mandatory
Valid Values	[true | **false**]
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI BATCH MOB
Installation Mode	EMBED NATIVE

WEB_JMX_RMI_PORT_PERFORMANCE - JMX Port for Web Application Server monitoring

Parameter	WEB_JMX_RMI_PORT_PERFORMANCE

Description	Port number for JMX monitoring for Online monitoring.
Ordinality	Mandatory
Valid Values	Valid open port number for monitoring.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_MAXAGE - Text Cache

Parameter	WEB_MAXAGE
Description	Set the duration (in seconds) to store page structures from the product, in the client cache, on the client browser.
Ordinality	Mandatory
Valid Values	See maxAge for valid values.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_MAXAGEI - Image Cache

Parameter	WEB_MAXAGEI
Description	Set the duration (in seconds) to store images, in the client cache, from the product on the client browser.
Ordinality	Mandatory
Valid Values	See maxAgeI for valid values.
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_MEMORY_OPT_MAX - Maximum Web Application Server JVM Memory

Parameter	WEB_MEMORY_OPT_MAX
Description	Maximum Web Application Server JVM Memory in MB. Corresponds to Xmx java option. In native mode implementations, this setting is set on the Server definition.
Ordinality	Optional
Valid Values	Valid Memory setting (in MB). Default: **1024**

J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED

WEB_MEMORY_OPT_MAXPERMSIZE - Web Application Server JVM Max PermGen size

Parameter	WEB_MEMORY_OPT_MAXPERMSIZE
Description	Maximum Web Application Server JVM PermGen Memory in MB. Corresponds to XX:Permsize java option. This options does not apply to Oracle JRockit implementations. In native mode implementations, this setting is set on the Server definition.
Ordinality	Optional
Valid Values	Valid Memory setting (in MB). Default: **500**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED

WEB_MEMORY_OPT_MIN - Initial Web Application Server JVM Memory

Parameter	WEB_MEMORY_OPT_MIN
Description	Initial Web Application Server JVM Memory in MB. Corresponds to Xms java option. In native mode implementations, this setting is set on the Server definition.
Ordinality	Optional
Valid Values	Valid Memory setting (in MB). Default: **1024**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED

WEB_NODENAME - IBM WebSphere ND Node Name

Parameter	WEB_NODENAME
Description	Node Name for IBM WebSphere ND.
Ordinality	Mandatory
Valid Values	Valid IBM WebSphere ND node name
J2EE Web Application Server	WASND

Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED NATIVE

WEB_PRELOADALL - Preload all pages On startup

Parameter	WEB_PRELOADALL
Description	Enable or Disable preload of pages at startup. This determines the scope of any reload. A value of **false** (default) only preloads the Main menu only. A value of **true** preloads all functions on all menus. A value of **true** will lengthen the startup process.
Ordinality	Mandatory
Valid Values	[true|false]
J2EE Web Application Server	WLS WAS WASND
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_PRINCIPAL_NAME - Default J2EE Authorization Principal

Parameter	WEB_PRINCIPAL_NAME
Description	Principal Name for online application.
Ordinality	Mandatory
Valid Values	Valid Principal Name. Default: **cisusers**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI
Installation Mode	EMBED NATIVE

WEB_ROLE_NAME - Default J2EE Authorization Role

Parameter	WEB_ROLE_NAME
Description	Role Name for online application
Ordinality	Mandatory
Valid Values	Valid Role Name. Default: **cisusers**
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI
Installation Mode	EMBED NATIVE

WEB_SERVER_HOME - Location Of Web Application Server

Parameter	WEB_SERVER_HOME
Description	Generic location of Web Application Server. This value is then used to set the following values: WAS_HOME - IBM WebSphere WASND_HOME - IBM WebSphere ND WL_HOME - Oracle WebLogic
Ordinality	Mandatory
Valid Values	Home location of Web Application Server software
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI
Installation Mode	EMBED NATIVE

WEBSERVICES_ENABLED - Whether Inbound Web Services Is Enabled

Parameter	WEBSERVICES_ENABLED
Description	Enable or Disable Inbound Web Services support.
Ordinality	Mandatory
Valid Values	[true|**false**]
J2EE Web Application Server	WLS WAS WASND
Tier	IWS
Installation Mode	EMBED NATIVE

WEB_SPLPASS - Application Administration Password

Parameter	WEB_SPLPASS
Description	Encrpyted password for administration user for Oracle WebLogic. This setting is only applicable to Oracle WebLogic customers and is only used for embedded mode installations.
Ordinality	Mandatory
Valid Values	Encrypted password for administration user.
J2EE Web Application Server	WLS
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_SPLUSER - Application Administration Userid

Parameter	WEB_SPLUSER
Description	Administration user for Oracle WebLogic. This setting is only applicable to Oracle WebLogic customers and is only used for embedded mode installations.
Ordinality	Mandatory
Valid Values	Administration user. Default: **system**
J2EE Web Application Server	WLS
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_SVRNAME - IBM WebSphere Server Name

Parameter	WEB_SVRNAME
Description	Server Name for IBM WebSphere/IBM WebSphere ND.
Ordinality	Mandatory
Valid Values	Valid IBM WebSphere/IBM WebSphere ND server name
J2EE Web Application Server	WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED NATIVE

WEB_WASPASS - IBM WebSphere JNDI Password

Parameter	WEB_WASPASS
Description	Encrpyted password for JNDI user for IBM WebSphere/IBM WebSphere ND. This setting is only applicable to IBM WebSphere/IBM WebSphere ND customers.
Ordinality	Mandatory
Valid Values	Encrypted password for JNDI user.
J2EE Web Application Server	WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_WASUSER- IBM WebSphere JNDI Userid

Parameter	**WEB_WASUSER**
Description	Administration JNDI user for IBM WebSphere/IBM WebSphere ND. This parameter is used for any JNDI calls to the servers. This setting is only applicable to IBM WebSphere/IBM WebSphere ND customers.
Ordinality	Mandatory
Valid Values	JNDI Administration User.
J2EE Web Application Server	WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_WLAUTHMETHOD - Authentication Method

Parameter	**WEB_WLAUTHMETHOD**
Description	J2EE Authorization Method implemented.
Ordinality	Mandatory
Valid Values	Valid Values: • **FORM** (Default) - Use login forms • **BASIC** - Use Operating System Login • **CLIENT-CERT** - Client Certificate
J2EE Web Application Server	WLS WAS WASND
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED NATIVE

WEB_WLHOST - Web Server Host

Parameter	**WEB_WLHOST**
Description	Host Name for Oracle WebLogic Install. This is used for Oracle WebLogic customers only and embedded installations only.
Ordinality	Mandatory
Valid Values	Valid Host Name. Default: **localhost**
J2EE Web Application Server	WLS
Tier	WEB BAS IWS XAI MOB
Installation Mode	EMBED

WEB_WLPAGECHECKSECONDS - Interval for recompilation of JSP

Parameter	**WEB_WLPAGECHECKSECONDS**
Description	Interval, in seconds, to force recompilation for server JSP's. This is used for Oracle WebLogic customers only. This setting is designed to be altered for development environments. The default is recommended for all other environments.
Ordinality	Mandatory
Valid Values	Valid Recompile time. Default: **43200**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_WLPORT - Web Server HTTP Port

Parameter	**WEB_WLPORT**
Description	Port number for environment for online system.
Ordinality	Mandatory
Valid Values	Valid open port number. Default: **6500**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_WLSSLPORT - Oracle WebLogic SSL HTTP Port.

Parameter	**WEB_WLSSLPORT**
Description	SSL Port number for environment for online system.
Ordinality	Mandatory
Valid Values	Valid open port number. Default: **6501**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_WLSYSPASS - Oracle WebLogic JNDI System Password

Parameter	WEB_WLSSYSPASS
Description	Encrypted Oracle WebLogic JNDI Password used for JNDI lookups. This is used for Oracle WebLogic customers only.
Ordinality	Mandatory
Valid Values	Valid JNDI Password.
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_WLSYSUSER - Oracle WebLogic JNDI System Userid

Parameter	WEB_WLSSYSUSER
Description	Oracle WebLogic JNDI User used for JNDI lookups. This is used for Oracle WebLogic customers only.
Ordinality	Mandatory
Valid Values	Valid JNDI User.
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED NATIVE

WEB_WLS_SVRNAME - Oracle WebLogic Server Name

Parameter	WEB_WLS_SVRNAME
Description	Oracle WebLogic Server Name. This is used for Oracle WebLogic customers only and embedded installations only.
Ordinality	Mandatory
Valid Values	Valid Server Name. Default: **myserver**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

WLS_ADMIN_PORT - Admin Console Port Number

Parameter	WLS_ADMIN_PORT
Description	Oracle WebLogic Administration Port. Only supplied if Administration port differs from application port. Oracle

	WebLogic Server Name. This is used for Oracle WebLogic customers only and embedded installations only.
Ordinality	Mandatory
Valid Values	Valid Server Name. Default: **myserver**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

WLS_WEB_WLSYSPASS - Console Password for Oracle WebLogic

Parameter	WLS_WEB_WLSYSPASS
Description	Encrypted Oracle WebLogic Password used for console. This is used for Oracle WebLogic customers only and embedded installations only.
Ordinality	Mandatory
Valid Values	Valid console Password.
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

WLS_WEB_WLSYSUSER - Console User for Oracle WebLogic

Parameter	WLS_WEB_WLSYSUSER
Description	Encrypted Oracle WebLogic user used for console. This is used for Oracle WebLogic customers only and embedded installations only.
Ordinality	Mandatory
Valid Values	Valid console user. Default: **system**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

WL_HOME - Oracle WebLogic Home

Parameter	WL_HOME
Description	Home directory of Oracle WebLogic. . This is used for Oracle WebLogic customers only and embedded installations only.

Ordinality	Mandatory
Valid Values	Valid console user. Default: **system**
J2EE Web Application Server	WLS
Tier	WEB
Installation Mode	EMBED

XAIApp - Name of XAI WAR file

Parameter	XAIApp
Description	Name of XAI WAR file as generated for builds. This is generated and should not be changed.
Ordinality	Mandatory
Valid Values	Valid WAR file name. Default: **XAIApp.war**
J2EE Web Application Server	WLS WAS WASND
Tier	XAI
Installation Mode	EMBED NATIVE

XAI_DBPASS - Password for Database User for XAI

Parameter	XAI_DBPASS
Description	Encrypted database user password for the XAI component of the product.
Ordinality	Mandatory
Valid Values	Encrypted Password
J2EE Web Application Server	WLS WAS WASND
Tier	XAI DB
Installation Mode	EMBED NATIVE

XAI_DBUSER - Database User used for XAI

Parameter	XAI_DBUSER
Description	Database User, with Read/Write permissions used for pooled connections for XAI transactions.
Ordinality	Mandatory
Valid Values	Valid Database User used in installation
J2EE Web Application Server	WLS WAS WASND
Tier	XAI DB

Installation Mode	EMBED NATIVE

Web.xml

The Web deployment descriptor editor lets you specify deployment information for modules created in the Web development environment. The information appears in the *web.xml* file. The *web.xml* file for a Web project provides information necessary for deploying a Web application module. It is used in building a WAR/EAR file from a project.

The Business Application is controlled by a configuration file that holds behavioral information for the applications. Refer to http://jcp.org/en/jsr/detail?id=109 for more details of the format. For example:

```
...
   <env-entry>
      <description>How long to cache drop down values in
seconds</description>
      <env-entry-name>fieldValuesAge</env-entry-name>
      <env-entry-value>3600</env-entry-value>
      <env-entry-type>java.lang.Integer</env-entry-type>
   </env-entry>
...
```

disableCompression - Disable Compression

Parameter	**disableCompression**
Description	Enables or disables compression between the Web Application Server and Business Application Server (true or false).
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **WEB_ISDEVELOPMENT** parameter from **ENVIRON.INI**. It is highly recommended to set to false for all environment exception is development environments using Oracle Utilities SDK.
Applicability	WEB

Example:	`<context-param>`
	`<param-name>`**disableCompression**`</param-name>`
	`<param-value>`**false**`</param-value>`
	`</context-param>`

maxAgeI - Max Age for Images

Parameter	**maxAge**
Description	Set the duration (in seconds) to store images, in the client cache,

	from the product on the client browser.
Ordinality	Mandatory
Valid Values	As per J2EE Standards. Default: **28850**
Source	Derived from **WEB_MAXAGEI** parameter from **ENVIRON.INI**. It is highly recommended to set to the default unless otherwise recommended.
Applicability	WEB
Example:	

```
<filter>
    <filter-name>Image Browser Cache Filter</filter-name>
    <filter-
class>com.splwg.base.web.services.BrowserCacheFilter</filter-
class>
     <init-param>
         <param-name>maxAge</param-name>
         <param-value>28850</param-value>
     </init-param>
</filter>
```

maxAge - Max Age for pages

Parameter	maxAge
Description	Set the duration (in seconds) to store page structures from the product, in the client cache, on the client browser.
Ordinality	Mandatory
Valid Values	As per J2EE Standards. Default: **28850**
Source	Derived from **WEB_MAXAGE** parameter from **ENVIRON.INI**. It is highly recommended to set to the default unless otherwise recommended.
Applicability	WEB
Example:	

```
<filter>
    <filter-name>Browser Cache Filter</filter-name>
    <filter-
class>com.splwg.base.web.services.BrowserCacheFilter</filter-
class>
     <init-param>
         <param-name>maxAge</param-name>
         <param-value>28850</param-value>
     </init-param>
</filter>
```

auth-method - Authorization Method

Parameter	auth-method

Description	Sets the J2EE authentication method for the product.
Ordinality	Mandatory
Valid Values	[BASIC|CLIENT-CERT|**FORM**]
Source	Derived from **WEB_WLAUTHMETHOD** parameter from **ENVIRON.INI**. Depending on the value this may lead to further settings being implemented. • **BASIC** - Enables the operating system to provide the logon dialog. • **CLIENT-CERT** - Enables an external solution such as a SSO solution, Kerberos or any security implementing client certificates to provide credentials. • **FORM** - This instructs the product to implement the following additional settings: ▪ **WEB_FORM_LOGIN_PAGE** and **WEB_HELP_FORM_LOGIN_ERROR_PAGE** need to be set to indicate the form to use for online authentication. ▪ **WEB_HELP_FORM_LOGIN_PAGE** and **WEB_HELP_FORM_LOGIN_ERROR_PAGE** need to be set to indicate the form to use for online help authentication. ▪ **WEB_APPVIEWER_FORM_LOGIN_PAGE** and **WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE** need to be set to indicate the form to use for AppViewer authentication.
Applicability	WEB

Example:	`<login-config>`
	`<auth-method>FORM</auth-method>`
	`<form-login-config>`
	`<form-login-page>/loginPage.jsp</form-login-page>`
	`<form-error-page>/formLoginError.jsp</form-error-page>`
	`</form-login-config>`
	`</login-config>`

fieldValuesAge - Server Cache Lifetime

Parameter	fieldValuesAge
Description	Set the duration (in seconds) to hold data in the server data cache.
Ordinality	Mandatory
Valid Values	1 - 2147483647 Default: **3600**

Source	Defaulted from template. Changes to the value require a custom template.
Applicability	WEB BAS
Example:	

```
<env-entry>
  <description>How long to cache drop down values in seconds</description>
   <env-entry-name>fieldValuesAge</env-entry-name>
   <env-entry-type>java.lang.Integer</env-entry-type>
   <env-entry-value>3600</env-entry-value>
</env-entry>
```

disableUIPageCompression - Disable UI Compression

Parameter	disableUIPageCompression
Description	Enables or disables compression between the browser and Web Application Server (true or false).
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **WEB_ISDEVELOPMENT** parameter from **ENVIRON.INI**. It is highly recommended to set to **false** for all environment exception is development environments using Oracle Utilities SDK.
Applicability	WEB
Example:	

```
<env-entry>
  <description>Disable UIPage compression</description>
  <env-entry-name>disableUIPageCompression</env-entry-name>
  <env-entry-type>java.lang.Boolean</env-entry-type>
  <env-entry-value>false</env-entry-value>
</env-entry>
```

spl.properties

The **spl.properties** file is used by each layer of the product to configure the technical aspects of the product behavior. All settings in this file are of the format:

<parameter>=<parameter value>

com.oracle.ouaf.system.keystore.alias - Keystore Alias

Parameter	com.oracle.ouaf.system.keystore.alias
Description	Sets name alias in keystore. Corresponds to the **alias** option on **keytool**.
Ordinality	Mandatory
Valid Values	Set to **ouaf.system**.

Source	This value is et by the **KS_ALIAS** value in **ENVIRON.INI**.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.alias=ouaf.system

com.oracle.ouaf.system.keystore.hmac_key_alias - HMAC alias

Parameter	**com.oracle.ouaf.system.keystore.hmac_key_alias**
Description	Sets HMAC name alias in keystore. Corresponds to the **alias** option on **keytool**.
Ordinality	Mandatory
Valid Values	Set to **ouaf.system.hmac** by default.
Source	This value is set by the **KS_HMAC_ALIAS** value in **ENVIRON.INI**.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.hmac_key_alias=ouaf.system.hmac

com.oracle.ouaf.system.keystore.file - Location Of Keystore

Parameter	**com.oracle.ouaf.system.keystore.file**
Description	Location of keystore file. Corresponds to the **keystore** option on **keytool**. This entry should not be altered.
Ordinality	Mandatory
Valid Values	Set to full path location of keystore and name of keystore file.
Source	Defaulted to **$SPLEBASE/ks/.ouaf_keystore** (or for Windows **%SPLEBASE%\ks\.ouaf_keystore**) from the **KS_KEYSTORE_FILE** value in **ENVIRON.INI**.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.alias=ouaf.system

com.oracle.ouaf.system.keystore.mode - Keystore Mode

Parameter	**com.oracle.ouaf.system.keystore.mode**
Description	Mode of the keystore. This is not used to create the keystore, just how the product processes the keys inside.
Ordinality	Mandatory
Valid Values	Set to valid keystore mode. Defaulted to **CBC**.
Source	This value is set from the **KS_MODE** value in **ENVIRON.INI**.
Applicability	WEB BAS IWS XAI BATCH MOB

Example: com.oracle.ouaf.system.keystore.mode=CBC

com.oracle.ouaf.system.keystore.padding - Key Padding

Parameter	com.oracle.ouaf.system.keystore.padding
Description	Padding used for keys. This is not used to create the keystore, just how the product processes the keys inside.
Ordinality	Mandatory
Valid Values	Set to valid keystore mode. Defaulted to PKCS5Padding.
Source	This value is set from the KS_PADDING value in ENVIRON.INI.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.padding=PKCS5Padding

com.oracle.ouaf.system.keystore.passwordFileName - Password File

Parameter	com.oracle.ouaf.system.keystore.passwordFileName
Description	Location of keystore password file. Corresponds to the storepass option on keytool. This entry should not be altered.
Ordinality	Mandatory
Valid Values	Set to full path location of keystore and name of keystore file.
Source	Defaulted to $SPLEBASE/ks/.ouaf_storepass (or for Windows %SPLEBASE%\ks\.ouaf_storepass) from the KS_STOREPASS_FILE value in ENVIRON.INI.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.passwordFileName= /spl/OUAFDEMO/ks/.ouaf_storepass

com.oracle.ouaf.system.keystore.type - Keystore Type

Parameter	com.oracle.ouaf.system.keystore.type
Description	Keystore Type. Corresponds to the storetype option on keytool.
Ordinality	Mandatory
Valid Values	Set to valid keystore mode. Defaulted to JCEKS.
Source	This value is set from the KS_STORETYPE value in ENVIRON.INI.
Applicability	WEB BAS IWS XAI BATCH MOB
Example:	com.oracle.ouaf.system.keystore.type=JCEKS

com.oracle.XPath.LRUSize - XPath Cache

Parameter	com.oracle.XPath.LRUSize
Description	Sets Maximum XPath queries cached used in Schema based objects.
Ordinality	Optional
Valid Values	0 - No queries cached, **-1** - Unlimited (7k per query), Value - Number of queries.
Source	Defaulted from templates. Use user exit spl.properties.service.exit.include to implement a custom value.
Applicability	BAS
Example:	com.oracle.XPath.LRUSize=2000

com.oracle.XPath.flushTimeout - XPath Flush

Parameter	com.oracle.XPath.flushTimeout
Description	Sets time (in seconds) to reset XPath cache.
Ordinality	Optional
Valid Values	**0** - Never auto flush, *Value* - Time in seconds between flushes.
Source	Defaulted from templates. Use user exit spl.properties.service.exit.include to implement a custom value.
Applicability	BAS
Example:	com.oracle.XPath.flushTimeout=86400

com.splwg.batch.cluster.jvmName - Online Batch JVM Name

Parameter	com.splwg.batch.cluster.jvmName
Description	Sets name for monitoring purposes for online batch JVM for online submission. This value is not used by the runtime, just exposed to the JMX interface for categorization purposes.
Ordinality	Optional
Valid Values	Any string without embedded blanks.
Source	Use the spl.properties.root.exit.include user exit to set this value.
Applicability	WEB
Example:	com.splwg.batch.cluster.jvmName=onlineBatch

com.splwg.batch.scheduler.daemon - Online Submission Daemon

Parameter	com.splwg.batch.scheduler.daemon
Description	Enable or Disable the Batch Daemon for online submission. The Batch Daemon is used to detect *Pending* Batch Submission objects and send them to Batch Online servers. It is highly recommended to specify a value of **false** for Production environments. The Batch Daemon is only recommended to be used, set to **true**, in Testing, Development or Demonstration environments.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **BATCHDAEMON** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	com.splwg.batch.scheduler.daemon=false

com.splwg.grid.executionMode - Online Execution Mode

Parameter	com.splwg.grid.executionMode
Description	Sets the execution mode of the online Batch submission.
Ordinality	Mandatory
Valid Values	[DISTRIBUTED|**CLUSTERED**]
Source	Derived from **BATCH_MODE** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	com.splwg.grid.executionMode=CLUSTERED

com.splwg.grid.online.enabled - Online Batch Server

Parameter	com.splwg.grid.online.enabled
Description	Enable or Disable the Batch Server for online submission. The Batch Server is reponsible for managing and executing jobs allocated by the Batch Daemon. It is highly recommended to specify a value of **false** for Production environments. The Batch Server is only recommended to be used in Testing, Development or Demonstration environments.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **BATCHENABLED** parameter from **ENVIRON.INI**.
Applicability	BAS

Example: `com.splwg.grid.online.enabled=false`

com.splwg.grid.distThreadPool.threads.DEFAULT - Online Submission Threadpool

Parameter	`com.splwg.grid.distThreadPool.threads.DEFAULT`
Description	Maximum number of concurrent threads used to execute online submission. This setting is used in conjunction with the Online Batch Server setting.
Ordinality	Mandatory
Valid Values	1 - 1000. Default: **5**
Source	Derived from **BATCHTHREADS** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	`com.splwg.grid.distThreadPool.threads.DEFAULT=5`

com.splwg.schema.newValidations.F1 - Set behavior of validations

Parameter	`com.splwg.schema.newValidations.F1`
Description	Sets behavior of schema validations. Internal use only. This is set by individual products and should not be altered unless otherwise instructed by Oracle Support.
Ordinality	Mandatory
Valid Values	[**true**|false]
Source	Value inherited from templates. Use custom templates to set this value to **false**, if desired.
Applicability	WEB BAS
Example:	`com.splwg.schema.newValidations.F1=true`

jmx.remote.x.access.file - JMX Access Control

Parameter	`jmx.remote.x.access.file`
Description	Sets relative location of JMX access control file holding the valid user. Refer to JMX Security for more details of JMX security.
Ordinality	Mandatory
Valid Values	Relative location of file from **$SPLEBASE/%SPLEBASE%**. Defaults to **scripts/ouaf.jmx.access.file**
Source	Value inherited from templates. Use custom templates to set this value to the desired location and file, if desired.
Applicability	WEB BAS

Example: `jmx.remote.x.access.file=scripts/ouaf.jmx.access.file`

jmx.remote.x.password.file - JMX Password Control

Parameter	**`jmx.remote.x.password.file`**
Description	Sets relative location of JMX password file holding the valid passwords for **`jmx.remote.x.access.file`**. Refer to JMX Security for more details of JMX security.
Ordinality	Mandatory
Valid Values	Relative location of file from **`$SPLEBASE`**/**`%SPLEBASE%`**. Defaults to **`scripts/ouaf.jmx.password.file`**
Source	Value inherited from templates. Use custom templates to set this value to the desired location and file, if desired.
Applicability	WEB BAS
Example:	`jmx.remote.x.password.file=scripts/ouaf.jmx.password.file`

ouaf.accessiblity.features - Enable Accessibility features

Parameter	**`ouaf.accessiblity.features`**
Description	Enables or disabled accessibility support from the product, specifically around implementing tabIndex for column headers. Other accessibility features are not altered by this setting.
Ordinality	Mandatory
Valid Values	[**true**|false]
Source	Value inherited from templates. Use custom templates to set this value to **`false`**, if desired.
Applicability	WEB
Example:	`ouaf.accessiblity.features=true`

ouaf.runtime.compatibility.enablePrecisionScaleOnMoney - Money Processing

Parameter	**`ouaf.runtime.compatibility.enablePrecisionScaleOnMoney`**
Description	Enables or disabled precision checking on money fields.
Ordinality	Mandatory
Valid Values	[**true**|false]
Source	Value inherited from templates. Use custom templates to set this value to **`false`**, if desired.
Applicability	WEB
Example:	`ouaf.runtime.compatibility.enablePrecisionScaleOnMoney=false`

ouaf.jmx.com.oracle.ouaf.ws.mbeans.WSFlushBean - WS Cache Flush

Parameter	`ouaf.jmx.com.oracle.ouaf.ws.mbeans.WSFlushBean`
Description	Enables or disables the Flush mbean used for the online JMX monitoring. Refer to the Web Services JMX interface for more information. This parameter is only enabled if the JMX interface is enabled.
Ordinality	Mandatory
Valid Values	[**enabled**|disabled]
Source	Value inherited from templates. Use custom templates to set this value to `disabled`, if desired.
Applicability	IWS
Example:	`ouaf.jmx.com.oracle.ouaf.ws.mbeans.WSFlushBean=enabled`

ouaf.jmx.com.splwg.base.support.management.mbean.JVMInfo - JVM Information

Parameter	`ouaf.jmx.com.splwg.base.support.management.mbean.JVMInfo`
Description	Enables or disables the JVMInfo mbean used for the online JMX monitoring. Refer to the Web Application Server JMX interface for more information. This parameter is only enabled if the JMX interface is enabled.
Ordinality	Mandatory
Valid Values	[**enabled**|disabled]
Source	Value inherited from templates. Use custom templates to set this value to `disabled`, if desired.
Applicability	WEB
Example:	`ouaf.jmx.com.splwg.base.support.management.mbean.JVMInfo=enabled`

ouaf.jmx.com.splwg.base.web.mbeans.FlushBean - Manage Cache

Parameter	`ouaf.jmx.com.splwg.base.web.mbeans.FlushBean`
Description	Enables or disables the mbean to reset the online data cache manually. Refer to the Web Application Server JMX interface for more information. This parameter is only enabled if the JMX interface is enabled.
Ordinality	Mandatory
Valid Values	[**enabled**|disabled]
Source	Value inherited from templates. Use custom templates to set this value to `disabled`, if desired.
Applicability	WEB

Example: `ouaf.jmx.com.splwg.base.web.mbeans.FlushBean=enabled`

ouaf.messaging.transformerfactory - Tranformer

Parameter	**ouaf.messaging.transformerfactory**
Description	Sets the transformation factory for the real time adapters. This parameter is only set if requested by Oracle Support.
Ordinality	Optional
Valid Values	Set to one of the following values: com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl org.apache.xalan.xsltc.trax.TransformerFactoryImpl
Source	Value inherited from J2EE container. Use custom templates to set this value, if alternative desired
Applicability	BAS BATCH
Example:	`ouaf.messaging.transformerfactory = org.apache.xalan.xsltc.trax.TransformerFactoryImpl`

ouaf.shortcut.ignore.altKeys - Disable Hot Keys

Parameter	**ouaf.shortcut.ignore.altKeys**
Description	Allows specific hot keys to be disabled to support specific keyboard layouts (e.g. Polish keyboards).
Ordinality	Optional
Valid Values	Comma separated list of keys to disable
Source	Set within template. Use the **spl.properties.root.exit.include** user exit to set this value.
Applicability	WEB
Example:	`ouaf.shortcut.ignore.altKeys=C,L`

ouaf.timeout.business_object.default - Set Default Business Object Timeout

Parameter	**ouaf.timeout.business_object.default**
Description	Sets the default timeout (in seconds) for Business Objects. Refer to Online Transaction Timeouts for more information.
Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS

Example: `ouaf.timeout.business_object.default=10`

ouaf.timeout.business_object.<bocode> - Set Specific Business Object Timeout

Parameter: **ouaf.timeout.business_object.*<bocode>***

Description: Sets the timeout (in seconds) for Business Object designated by ***<bocode>*** which corresponds to the Business Object Name. This overrides the **ouaf.timeout.business_object.default** setting for the indicated business object. Refer to Online Transaction Timeouts for more information.

To find valid values for Business Objects, use the Business Object Maintenance function or use the following SQL:

```
SELECT bus_obj_cd, descr
FROM f1_bus_obj_l
where language_cd = 'ENG';
```

The **LANGUAGE_CD** in the above query can be altered to other languages for language packs installed.

Ordinality: Optional

Valid Values: 1 - 2147483647

Source: Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.

Applicability: BAS

Example: `ouaf.timeout.business_object.CMUser=10`

ouaf.timeout.business_service.default - Set Default Business Service Timeout

Parameter: **ouaf.timeout.business_service.default**

Description: Sets the default timeout (in seconds) for Business Services. Refer to Online Transaction Timeouts for more information.

Ordinality: Optional

Valid Values: 1 - 2147483647

Source: Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.

Applicability: BAS

Example: `ouaf.timeout.business_service.default=10`

ouaf.timeout.business_service.<bscode> - Set Specific Business Service Timeout

Parameter: **ouaf.timeout.business_service.*<bscode>***

Description	Sets the timeout (in seconds) for Business Service designated by ***<bscode>*** which corresponds to the Business Service Name. This overrides the **ouaf.timeout.business_service.default** setting for the indicated business service. Refer to Online Transaction Timeouts for more information.
	
	To find valid values for Business Services, use the Business Service Maintenance function or use the following SQL:
	`SELECT bus_svc_cd, descr`
	`FROM f1_bus_svc_l`
	`where language_cd = 'ENG';`
	
	The **LANGUAGE_CD** in the above query can be altered to other languages for language packs installed

Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.business_service.F1-EmailService=10`

ouaf.timeout.query.default - Set Default Query Timeout

Parameter	**ouaf.timeout.query.default**
Description	Sets the default timeout (in seconds) for query zones (searches and lists). Refer to Online Transaction Timeouts for more information. Use the Zone maintenance function to determine valid values. This setting only applies to the following zone types: **F1-DE**, **F1-DE-MULQRY**, **F1-DE-QUERY** and **F1-DE-SINGLE**
Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.query.default=10`

ouaf.timeout.query.<zonecode> - Set Specific Query Timeout

Parameter	**ouaf.timeout.query.*<zonecode>***
Description	Sets the timeout (in seconds) for the online query zone designated by ***<zonecode>*** which corresponds to the Query Zone. This overrides the

ouaf.timeout.query.default setting for the indicated query zone.

Refer to Online Transaction Timeouts for more information.

This setting only applies to the following zone types:

F1-DE, **F1-DE-MULQRY**, **F1-DE-QUERY** and **F1-DE-SINGLE**

Use the Zone maintenance function to determine valid values. Alternatively it is possible to use SQL to fine the ZONE_CD's *<zonecode>*:

```
SELECT z.zone_cd, d.descr,  z.zone_hdl_cd
     FROM ci_zone z, ci_zone_l d
     WHERE z.zone_hdl_cd LIKE 'F1-DE%'
     AND z.zone_cd     = d.zone_cd
     AND d.language_cd = 'ENG';
```

The **LANGUAGE_CD** in the above query can be altered to other languages for language packs installed.

Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.business_service.F1-MSTCFGLS=10`

ouaf.timeout.script.default - Set Default Script Timeout

Parameter	**ouaf.timeout.script.default**
Description	Sets the default timeout (in seconds) for Scripts. Refer to Online Transaction Timeouts for more information.
Ordinality	Optional
Valid Values	1 - 32767
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.script.default=10`

ouaf.timeout.script.<scriptname> - Set Specific Script Timeout

Parameter	**ouaf.timeout.script.*<scriptname>***
Description	Sets the timeout (in seconds) for Scripts designated by ***<scriptname>*** which corresponds to the Script Name. This overrides the **ouaf.timeout.script.default** setting for the indicated script. Refer

to Online Transaction Timeouts for more information.

Note: This timeout does not apply to plug-in scripts.

To find valid values for Script, use the Script Maintenance function or use the following SQL:

```
SELECT s.scr_cd, d.descr254
FROM ci_scr s, ci_scr_l d
where s.scr_cd = d.scr_cd
and s.scr_type_flg in ('BPSC','SVSC')
and d.language_cd = 'ENG';
```

The **LANGUAGE_CD** in the above query can be altered to other languages for language packs installed

Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.script.F1-ViewWSDL=10`

ouaf.timeout.service.default - Set Default Service Timeout

Parameter	**ouaf.timeout.service.default**
Description	Sets the default timeout (in seconds) for Application Services. Refer to Online Transaction Timeouts for more information.
Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.service.default=10`

ouaf.timeout.service.<service> - Set Specific Service Timeout

Parameter	**ouaf.timeout.service.*<service>***
Description	Sets the timeout (in seconds) for Application Services designated by ***<service>*** which corresponds to the Application Service Name. This overrides the **ouaf.timeout.service.default** setting for the indicated Application Service. Refer to Online Transaction Timeouts for more information. To find valid values for Application Service, use the Application Service

	Maintenance function or use the following SQL:
	`SELECT SVC_NAME, DESCR`
	`from CI_MD_SVC_L`
	`where language_cd = 'ENG';`
	The **LANGUAGE_CD** in the above query can be altered to other languages for language packs installed

Ordinality	Optional
Valid Values	1 - 2147483647
Source	Use the **spl.properties.service.timeouts.exit.include** user exit to set this value as outlined in Online Transaction Timeouts.
Applicability	BAS
Example:	`ouaf.timeout.service.CILTUSEP=10`

ouaf.web.appViewer.coreAppURL - AppViewer callback URL

Parameter	**ouaf.web.appViewer.coreAppURL**
Description	URL for AppViewer to navigate back to the product.
Ordinality	Mandatory
Valid Values	Generated URL from **WL_HOST**, **WEB_WLPORT**, **WEB_WLSSLPORT** and **WEB_CONTEXT_ROOT**.
Source	Value inherited from templates. Use custom templates to set this value to desired value.
Applicability	WEB
Example:	`ouaf.web.appViewer.coreAppURL=` `http://localhost:6500/ouaf/loginPage.jsp#fromAppViewer`

ouaf.web.disablePreload - Disable Screen Preload

Parameter	**ouaf.web.disablePreload**
Description	Enables or sisables screen preloading at startup. Screens that are preloaded are autoloaded at startup rather than at first use, saving time at runtime for first use transactions. This parameter works in conjunction with **ouaf.web.preloadAllPages**. It is highly recommended to set this value to **false** (default) for Production Environments. It is only suggested to set this to **true** in development environments that are used in conjunction with the Oracle Utilities SDK.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **WEB_ISDEVELOPMENT** parameter from **ENVIRON.INI**. It is

	highly recommended to set to the default unless otherwise recommended
Applicability	WEB
Example:	ouaf.web.help.coreAppURL= http://localhost:6500/ouaf/loginPage.jsp

ouaf.web.help.coreAppURL - Help callback URL

Parameter	ouaf.web.help.coreAppURL
Description	URL for Help to navigate back to the product.
Ordinality	Mandatory
Valid Values	Generated URL from **WL_HOST**, **WEB_WLPORT**, **WEB_WLSSLPORT** and **WEB_CONTEXT_ROOT**.
Source	Value inherited from templates. Use custom templates to set this value to desired value.
Applicability	WEB
Example:	ouaf.web.help.coreAppURL= http://localhost:6500/ouaf/loginPage.jsp

ouaf.web.helpRootURL - Help URL

Parameter	ouaf.web.helpRootURL
Description	URL for Help for help requests from online.
Ordinality	Mandatory
Valid Values	Generated URL from and **WEB_CONTEXT_ROOT**.
Source	Value inherited from templates. Use custom templates to set this value to desired value.
Applicability	WEB
Example:	ouaf.web.helpRootURL=/ouaf/help/

ouaf.web.JspHeader.maxAge - JSP Age

Parameter	ouaf.web.JspHeader.maxAge
Description	Set the duration (in seconds) to screen HTML, in the client cache, from the product on the client browser.
Ordinality	Mandatory
Valid Values	As per J2EE Standards. Default: **28850**
Source	Derived from **WEB_MAXAGE** parameter from **ENVIRON.INI**. It is highly recommended to set to the default unless otherwise recommended.
Applicability	WEB

Example: `ouaf.web.JspHeader.maxAge=28850`

ouaf.web.preloadAllPages - Preload Scope

Parameter: **ouaf.web.preloadAllPages**

Description: Set the scope of the preloading of pages. This parameter is only used if the **ouaf.web.disablePreload** is set to **false**.

It is highly recommended to set this value to **false** (default) for all environments.

Ordinality: Mandatory

Valid Values: [true|**false**] - Main menu only (**false**), Main menu and Administration menu (true)

Source: Derived from **WEB_PRELOADALL** parameter from **ENVIRON.INI**. It is highly recommended to set to the default unless otherwise recommended.

Applicability: WEB

Example: `ouaf.web.preloadAllPages=false`

ouaf.ws.deploy.user - Web Services Deployment User

Parameter: **ouaf.ws.deploy.user**

Description: J2EE Administration account used to deploy Inbound Web Services container. Password for this user is contained in keystore as outlined in Enabling Inbound Web Services.

Ordinality: Mandatory

Valid Values: Valid Administration account for J2EE container.

Source: Derived from **WLS_WEB_WLSYSUSER** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.

Applicability: IWS

Example: `ouaf.ws.deploy.user=system`

ouaf.ws.superusers - Effective Web Services Users

Parameter: **ouaf.ws.superusers**

Description: Sets the effective users for Inbound Web Services. This allows credentials from foreign systems to use credentials from the product.

This is only set to alternatives if effective users are used.

Ordinality: Mandatory

Valid Values: Valid comma separated list of credentials

Source	Derived from **WEB_SPLUSER** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	`ouaf.ws.superusers=SYSUSER`

ouaf.ws.tools.artifact.generated.buildfile - Build File

Parameter	**ouaf.ws.tools.artifact.generated.buildfile**
Description	Location and name of ant build file used to generate the WAR file used for Inbound Web Services. This setting is used by the **iwsdeploy.sh** command as well as the online Inbound Web Services deployment tool.
Ordinality	Mandatory
Valid Values	Location and name of build file. Defaults to **$SPLEBASE/splapp/iws/iws-build.xml** (or **%$SPLEBASE%\splapp\iws\iws-build.xml** on Windows)
Source	Derived from **SPLEBASE** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	`ouaf.ws.tools.artifact.generated.buildfile= /spl/OUAFDEMO/splapp/iws/iws-build.xml`

ouaf.ws.tools.artifact.generated.sourcedir - Location of generated source

Parameter	**ouaf.ws.tools.artifact.generated.sourcedir**
Description	Location of the source generated for Inbound Web Services during the Web Services deployment process.
Ordinality	Mandatory
Valid Values	Location of generated source directory. Defaults to **$SPLEBASE/splapp/iws/gen** (or **%$SPLEBASE%\splapp\iws\gen** on Windows)
Source	Derived from **SPLEBASE** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	`ouaf.ws.tools.artifact.generated.sourcedir= /spl/OUAFDEMO/splapp/iws/gen`

ouaf.ws.tools.artifact.resource.dir - Location of resources

Parameter	**ouaf.ws.tools.artifact.resource.dir**
Description	Location of the resource files used in the generation of Inbound Web Services during the Web Services deployment process.

Ordinality	Mandatory
Valid Values	Location of resources used for Inbound Web Services. Defaults to **$SPLEBASE/splapp/iws/resources** (or **%$SPLEBASE%\splapp\iws\resources** on Windows)
Source	Derived from **SPLEBASE** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	ouaf.ws.tools.artifact.resource.dir= /spl/OUAFDEMO/splapp/iws/resources

ouaf.ws.tools.artifact.schema.dir - Location of Schemas

Parameter	**ouaf.ws.tools.artifact.schema.dir**
Description	Location of the schema files used in the generation of Inbound Web Services during the Web Services deployment process.
Ordinality	Mandatory
Valid Values	Location of schema files used for Inbound Web Services. Defaults to **$SPLEBASE/splapp/xai/schemas** (or **%$SPLEBASE%\splapp\xai\schemas** on Windows)
Source	Derived from **SPLEBASE** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	ouaf.ws.tools.artifact.schema.dir= /spl/OUAFDEMO/splapp/xai/schemas

ouaf.ws.tools.artifact.xsl.dir - Web Service XSL Location

Parameter	**ouaf.ws.tools.artifact.xsl.dir**
Description	Location of the XSLT files used by Inbound Web Services during the Web Services deployment process. This parameter can be overridden using the **ouaf.ws.xslDirOverride** setting.
Ordinality	Mandatory
Valid Values	Location of XSLT files used for Inbound Web Services. Defaults to **$SPLEBASE/splapp/xai/schemas** (or **%$SPLEBASE%\splapp\xai\schemas** on Windows). By default this value is shared with **ouaf.ws.tools.artifact.schema.dir**.
Source	Derived from **SPLEBASE** parameter from **ENVIRON.INI**. Use custom templates to set this value to an alternative desired value.

Applicability	IWS
Example:	`ouaf.ws.tools.artifact.xsl.dir=` `/spl/OUAFDEMO/splapp/xai/schemas`

ouaf.ws.tracing - Inbound Web Services Tracing

Parameter	**ouaf.ws.tracing**
Description	Enable or Disable tracing for Inbound Web Services.
Ordinality	Mandatory
Valid Values	[**true**|false]
Source	Defaulted from templates. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Example:	`ouaf.ws.tracing=true`

ouaf.ws.xslDirOverride - Override XSL Location

Parameter	**ouaf.ws.xslDirOverride**
Description	Location of the XSLT files used by Inbound Web Services during the Web Services deployment process. This parameter overrides the **ouaf.ws.tools.artifact.xsl.dir** setting.
Ordinality	Mandatory
Valid Values	Valid directories on server containing xsl For Windows environments, use the **file** protocol.
Source	Not set. Use custom templates to set this value to an alternative desired value.
Applicability	IWS
Examples:	`ouaf.ws.xslDirOverride=/temp` `ouaf.ws.xslDirOverride=file:///c:/temp`

ouaf.xai.HTTPBasicAuthPasswordEnc - Default XAI Password

Parameter	**ouaf.xai.HTTPBasicAuthPasswordEnc**
Description	Encrypted password for default application user for XML Application Integration.
Ordinality	Mandatory
Valid Values	Encrypted password for users.
Source	Derived from **WEB_SPLPASS** parameter from **ENVIRON.INI**.

Applicability	WEB
Example:	ouaf.xai.HTTPBasicAuthPasswordEnc=ENC(unjOoM7+zufXI5r6Q…)

ouaf.xai.HTTPBasicAuthUser - Default XAI User

Parameter	ouaf.xai.HTTPBasicAuthUser
Description	Default application user for XML Application Integration.
Ordinality	Mandatory
Valid Values	Valid Application User
Source	Derived from **WEB_SPLPUSER** parameter from **ENVIRON.INI**.
Applicability	WEB
Example:	ouaf.xai.HTTPBasicAuthUser=SYSUSER

ouaf.xai.XAIServerURL - Deafult XAI URL

Parameter	ouaf.xai.XAIServerURL
Description	Default URL for XML Application Integration.
Ordinality	Mandatory
Valid Values	Generated URL for XAI Server from **WEB_WLHOST**, **WEB_WLPORT** and **WEB_CONTEXT_ROOT** from **ENVIRON.INI**.
Source	Use custom templates to set this value to an alternative, if desired.
Applicability	WEB
Example:	ouaf.xai.XAIServerURL= http://localhost:6500/ouaf/XAIApp/xaiserver

spl.csv.delimiter.useFromDisplayProfile - List CSV delimeter

Parameter	spl.csv.delimiter.useFromDisplayProfile
Description	Enables or disables the display of the delimiter for CSV upload/downloads from zones.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from template. Use custom templates to set this value to **true**, if desired.
Applicability	WEB
Example:	spl.csv.delimiter.useFromDisplayProfile=false

spl.ejbContainer.contextFactory - Context Factory for Business Application Server

Parameter	**spl.ejbContainer.contextFactory**
Description	Default J2EE Context Factory to connect tiers.
Ordinality	Mandatory
Valid Values	Valid connection factory: Oracle WebLogic: **weblogic.jndi.WLInitialContextFactory** IBM WebSphere/IBM WebSphere ND: **com.ibm.websphere.naming.WsnInitialContextFactory**
Source	Derived from template. Use custom templates to set this value to an alternative, if desired.
Applicability	WEB IWS
Example:	`spl.ejbContainer.contextFactory=` `weblogic.jndi.WLInitialContextFactory`

spl.ejbContainer.password - Password for Business Application Server

Parameter	**spl.ejbContainer.password**
Description	Default encrypted password for tier defined for Business Application Server connections.
Ordinality	Mandatory
Valid Values	Valid password encryptyed Administration user
Source	Oracle WebLogic: **WEB_WLSYSPASS** from **ENVIRON.INI** IBM WebSphere/IBM WebSphere ND: **WEB_WASPASS** from **ENVIRON.INI** Use custom templates to set this value to an alternative, if desired.
Applicability	WEB IWS
Example:	`spl.ejbContainer.password=ENC(Hi7/RFy…)`

spl.ejbContainer.url - Business Application Server URL

Parameter	**spl.ejbContainer.url**
Description	URL to connect to the Business Application Server from the Web Application Server or Inbound Web Services.
Ordinality	Mandatory
Valid Values	Valid URL for:

	Oracle WebLogic: **t3://<server>:<port>** or **t3s://<server>:<port>** IBM WebSphere/IBM WebSphere ND: **iiop://<server>:<port>**
Source	Derived from template. Use custom templates to set this value to an alternative, if desired.
Applicability	WEB IWS
Example:	`spl.ejbContainer.url = t3://localhost:6500`

spl.ejbContainer.user - Business Application Server user

Parameter	**spl.ejbContainer.user**
Description	Administration user for Business Application server to connect to from the Web Application Server and Inbound Web Services.
Ordinality	Mandatory
Valid Values	Valid Administration User on J2EE Server
Source	Oracle WebLogic: **WEB_WLSYSUSER** from **ENVIRON.INI** IBM WebSphere/IBM WebSphere ND: **WEB_WASUSER** from **ENVIRON.INI** Use custom templates to set this value to an alternative, if desired.
Applicability	WEB IWS
Example:	`spl.ejbContainer.user=system`

spl.geocodeDatasource.contextFactory - GIS Context Factory

Parameter	**spl.geocodeDatasource.contextFactory**
Description	Default J2EE Context Factory to connect to GIS. This option will only be enabled if **GIS** is set to **true** in the **ENVIRON.INI**.
Ordinality	Mandatory
Valid Values	Valid connection factory: Oracle WebLogic: **weblogic.jndi.WLInitialContextFactory** IBM WebSphere/IBM WebSphere ND: **com.ibm.websphere.naming.WsnInitialContextFactory**
Source	Derived from template. Use custom templates to set this value to an alternative, if desired.

Applicability	WEB BAS BATCH
Example:	`spl.geocodeDatasource.contextFactory=weblogic.jndi.WLInitialContextFactory`

spl.geocodeDatasource.password - GIS Connection Password

Parameter	**spl.geocodeDatasource.password**
Description	Encrpted password for GIS connection for **spl.geocodeDatasource.user**. This option will only be enabled if **GIS** is set to **true** in the **ENVIRON.INI**.
Ordinality	Mandatory
Valid Values	Encrypted password for GIS.
Source	Derived from **GIS_WLSYSPASS** from **ENVIRON.INI**
Applicability	WEB BAS BATCH
Example:	`spl.geocodeDatasource.password=ENC(Hi7/RFy…)`

spl.geocodeDatasource.url - GIS Data Source

Parameter	**spl.geocodeDatasource.url**
Description	JNDI name for GIS Data Source. This option will only be enabled if **GIS** is set to **true** in the **ENVIRON.INI**.
Ordinality	Mandatory
Valid Values	Valid JNDI path for GIS Data Source
Source	Derived from **GIS_WLSYSUSER** from **ENVIRON.INI**
Applicability	WEB BAS BATCH
Example:	`spl.geocodeDatasource.url=t3://localhost:7666/jdbc/gisdb`

spl.geocodeDatasource.user - GIS Connection User

Parameter	**spl.geocodeDatasource.user**
Description	Valid JNDI user for GIS connection. This option will only be enabled if **GIS** is set to **true** in the **ENVIRON.INI**.
Ordinality	Mandatory
Valid Values	Encrypted password for GIS.
Source	Derived from **GIS_WLSYSUSER** from **ENVIRON.INI**
Applicability	WEB BAS BATCH
Example:	`spl.geocodeDatasource.user=gisadmin`

spl.mwm.abr.contextFactory - ABR Context Factory

Parameter	**spl.mwm.abr.contextFactory**
Description	Default J2EE Context Factory to connect to Appointment Booking Reservation (ABR) component.
Ordinality	Mandatory
Valid Values	Valid connection factory: Oracle WebLogic: **weblogic.jndi.WLInitialContextFactory** IBM WebSphere/IBM WebSphere ND: **com.ibm.websphere.naming.WsnInitialContextFactory**
Source	Derived from **CONTEXTFACTORY** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.abr.contextFactory= weblogic.jndi.WLInitialContextFactory`

spl.mwm.abr.password - ABR Connection Password

Parameter	**spl.mwm.abr.password**
Description	Encrpted password for Appointment Booking Reservation (ABR) component connection for **spl.mwm.abr.user**.
Ordinality	Mandatory
Valid Values	Encrypted password for ABR.
Source	Derived from **WLS_PASSWORD** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.abr.password=ENC(Hi7/RFy…)`

spl.mwm.abr.timeout - Timeout for ABR

Parameter	**spl.mwm.abr.timeout**
Description	Connection timeout, in seconds, for Appointment Booking Reservation (ABR) component.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **120000**)
Source	Derived from **TIMEOUT** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.abr.timeout=120000`

spl.mwm.abr.url - ABR URL

Parameter	**spl.mwm.abr.url**
Description	URL to connect to the Appointment Booking Reservation (ABR) component.
Ordinality	Mandatory
Valid Values	Valid URL for: Oracle WebLogic: **t3://<server>:<port>** or **t3s://<server>:<port>** IBM WebSphere/IBM WebSphere ND: **iiop://<server>:<port>**
Source	Derived from **URL** from **ENVIRON.INI**
Applicability	MOB
Example:	spl.abr.url = t3://localhost:7751

spl.mwm.abr.userid - ABR Connection User

Parameter	**spl.mwm.abr.user**
Description	Valid JNDI user for Appointment Booking Reservation (ABR) component connection.
Ordinality	Mandatory
Valid Values	Encrypted connetion user for GIS.
Source	Derived from **WLS_USERID** from **ENVIRON.INI**
Applicability	MOB
Example:	spl.mwm.abr.userid=ENC(ho4tqPhP…)

spl.mwm.scheduler.abr.maxProcessingTime - Processing Time for ABR

Parameter	**spl.mwm.scheduler.abr. maxProcessingTime**
Description	Transaction timeout, in seconds, for each appointment booking processed by Appointment Booking Reservation (ABR) component.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **5**)
Source	Derived from **MAXPROCESSINGTIME** from **ENVIRON.INI**
Applicability	MOB
Example:	spl.mwm.scheduler.abr.maxProcessingTime=5

spl.mwm.scheduler.abr.minRequests - Initial Scheduler Pool Size

Parameter	**spl.mwm.scheduler.abr.minRequests**
Description	Initial request pool size for Appointment Booking Reservation (ABR) component.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **1**)
Source	Derived from **MINREQUESTS** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.abr.minRequests=1`

spl.mwm.scheduler.cleanse.interval - Scheduler Cache Clense

Parameter	**spl.mwm.scheduler.cleanse.interval**
Description	Prefresh time, in seconds, for scheduler cache with product.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **900**)
Source	Derived from **CLEANSE_INTERVAL** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.cleanse.interval=900`

spl.mwm.scheduler.connectionTimeout - Scheduler Connection Timeout

Parameter	**spl.mwm.scheduler.connectionTimeout**
Description	Connection timeout, in seconds, between product and scheduler.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **300000**)
Source	Derived from **SCHED_CONN_TIMEOUT** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.connectionTimeout=300000`

spl.mwm.scheduler.maintenanceCycleTime - Scheduler Integrity Check

Parameter	**spl.mwm.scheduler.maintenanceCycleTime**
Description	Time, in seconds, between scheduler integrity checks.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **300**)

Source	Derived from **SCHED_CYCLE_TIMEOUT** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.maintenanceCycleTime=300`

spl.mwm.scheduler.mapDir - Map Directory

Parameter	**spl.mwm.scheduler.mapDir**
Description	Location of Map files if external maps are used.
Ordinality	Mandatory
Valid Values	Valid Directory
Source	Derived from **MAPDIR** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.mapDir=/spl/splapp/OUAFDEMO/sploutput`

spl.mwm.scheduler.nodeId - Scheduler Node

Parameter	**spl.mwm.scheduler.nodeId**
Description	Node Identifier for Scheduler. This setting is used for backward compatibility only as Scheduler now dynamically determines nodes.
Ordinality	Mandatory
Valid Values	Valid Node Id (If used must match Scheduler object)
Source	Derived from **NODEID** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.mwm.scheduler.nodeId=MyNode`

spl.runtime.mwm.scheduler.ipcStartPort - IPC Start Port

Parameter	**spl.runtime.mwm.scheduler.ipcStartPort**
Description	Starting port number for Inter Process Communication (IPC) between the scheduler and products.
Ordinality	Mandatory
Valid Values	Valid port numbers. Scheduler allocate port numbers starting from this number.
Source	Derived from **IPCSTARTPORT** from **ENVIRON.INI**
Applicability	MOB
Example:	`spl.runtime.mwm.scheduler.ipcStartPort=7568`

spl.runtime.cobol.cobrcall - Whether product supports COBOL

Parameter	`spl.runtime.cobol.cobrcall`
Description	Enables or disables java to COBOL interface within product. This setting should only be set to **`true`** at installation time and only for products that support COBOL.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Set within templates.
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.cobrcall=false`

spl.runtime.cobol.sql.cache.maxTotalEntries - COBOL SQL Cache

Parameter	`spl.runtime.cobol.sql.cache.maxTotalEntries`
Description	Number of SQL's kept in cache for COBOL based extensions. The higher the value the higher the memory footprint needed for the Client JVM's and any Threadpool JVM's. This setting should only be set for products that support COBOL. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **1000**)
Source	Derived from templates. Use custom templates to set this value to an alternative, if desired
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.sql.cache.maxTotalEntries=1000`

spl.runtime.cobol.sql.cursoredCache.maxRows - COBOL Cursor Cache

Parameter	`spl.runtime.cobol.sql.cache.maxTotalEntries`
Description	Number of SQL's cursors in cache for COBOL based extensions per connection. This setting should only be set for products that support COBOL. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **10**)

Source	Derived from templates. Use custom templates to set this value to an alternative, if desired
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.sql.cursoredCache.maxRows=10`

spl.runtime.cobol.sql.disableQueryCache - COBOL Query Cache

Parameter	**`spl.runtime.cobol.sql.disableQueryCache`**
Description	Enable or Disable the SQL Query Cache for COBOL based objects. This setting should only be set for products that support COBOL. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from templates. Use custom templates to set this value to an alternative, if desired
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.sql.disableQueryCache=false`

spl.runtime.cobol.encoding - Java/COBOL Encoding

Parameter	**`spl.runtime.cobol.encoding`**
Description	Java encoding setting (**`java.lang`**) for correct language support. Refer to the Canonical Name for java.io API and java.lang API list for valid values. This setting should only be set for products that support COBOL. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	Valid Encoding Language. Default: **UTF8**
Source	Derived from **`JAVAENCODING`** parameter from **`ENVIRON.INI`**.
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.encoding=UTF8`

spl.runtime.cobol.remote.jvm - Whether Child JVM is enabled

Parameter	**`spl.runtime.cobol.remote.jvm`**
Description	Enable or Disable the Child JVM for COBOL based objects. This setting should only be set for products that support COBOL and needs

to be set to **true** for those products.

It is highly recommended not to alter this value unless otherwise advised by Oracle Support.

Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **RJVM** parameter from **ENVIRON.INI**.
Applicability	WEB BAS
Example:	`spl.runtime.cobol.remote.jvm=false`

spl.runtime.cobol.remote.jvmcommand - Child JVM command

Parameter	**spl.runtime.cobol.remote.jvmcommand**
Description	Java Command Line for the Child JVM. This setting should only be set for products that support COBOL.
Ordinality	Mandatory
Valid Values	Generated JVM command line
Source	Derived from **JVMCOMMAND** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	`spl.runtime.cobol.remote.jvmcommand=/usr/local/java…`

spl.runtime.cobol.remote.jvmcount - Child JVM Count

Parameter	**spl.runtime.cobol.remote.jvmcount**
Description	Number of Child JVM's to start. This setting should only be set for products that support COBOL.
Ordinality	Mandatory
Valid Values	1 - 32767 (Default:2)
Source	Derived from **BSN_JVMCOUNT** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	`spl.runtime.cobol.remote.jvmcount=2`

spl.runtime.cobol.remote.jvmoptions - Child JVM Options

Parameter	**spl.runtime.cobol.remote.jvmoptions**
Description	Additional Java options for the Child JVM. This setting should only be set for products that support COBOL.
Ordinality	Mandatory

Valid Values	Valid Additional Java Options
Source	Derived from **JVMCHILD_OPTIONS** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	`spl.runtime.cobol.remote.jvmoptions=-XX:-UseParallelGC`

spl.runtime.cobol.remote.rmiStartPort - Child JVM Ports

Parameter	**spl.runtime.cobol.remote.rmiStartPort**
Description	Starting Port Number for RMI communication between product and child VM. Port numbers must be free from this port number. This setting should only be set for products that support COBOL.
Ordinality	Mandatory
Valid Values	Valid Port Number
Source	Derived from **BSN_RMIPORT** parameter from **ENVIRON.INI**.
Applicability	BAS
Example:	`spl.runtime.cobol.remote.rmiStartPort=7022`

spl.runtime.cobol.sql.fetchSize - SQL Results Cache

Parameter	**spl.runtime.cobol.sql.fetchSize**
Description	Number of fetch buffers for SQL statements in COBOL. This setting should only be set for products that support COBOL. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **150**)
Source	Derived from templates. Use custom templates to set this value to an alternative, if desired
Applicability	WEB BAS BATCH
Example:	`spl.runtime.cobol.sql.fetchSize=150`

spl.runtime.compatibility.uiMapDisableInputValue - Map Defaults

Parameter	**spl.runtime.compatibility.uiMapDisableInputValue**
Description	Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.1 in relation to blanking out default values in screens. Setting this value to **true** will emulate Oracle Utilities Application Framework V2.1 behavior.

	This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.1 and wish to retain that behavior.
Ordinality	Optional
Valid Values	[true|**false**]
Source	Manually set. Use user exit **spl.properties.exit.include** to set this value to **true**, if desired
Applicability	WEB
Example:	spl.runtime.compatibility.uiMapDisableInputValue=false

spl.runtime.compatibility.uiMapDropdownSelectFirstValue - Dropdown defaults

Parameter	**spl.runtime.compatibility.uiMapDropdownSelectFirstValue**
Description	Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.1 in relation to defaulting to the first value in dropdownlist. Setting this value to **true** will emulate Oracle Utilities Application Framework V2.1 behavior. This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.1 and wish to retain that behavior.
Ordinality	Optional
Valid Values	[true|**false**]
Source	Manually set. Use user exit **spl.properties.exit.include** to set this value to **true**, if desired
Applicability	WEB
Example:	spl.runtime.compatibility.uiMapDropdownSelectFirstValue=false

spl.runtime.compatibility.uiMapDisableTitle - Screen Title Defaults

Parameter	**spl.runtime.compatibility.uiMapDisableTitle**
Description	Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.1 in relation to rendering screen titles. Setting this value to **true** will emulate Oracle Utilities Application Framework V2.1 behavior. This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.1 and wish to retain that behavior.
Ordinality	Optional
Valid Values	[true|**false**]
Source	Manually set. Use user exit **spl.properties.exit.include** to set this value to **true**, if desired

Applicability	WEB
Example:	`spl.runtime.compatibility.uiMapDisableTitle=false`

spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs - Map Ids

Parameter	**spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs**
Description	Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.x in relation to generating an unique idenitifer's for HTML for duplicate elements in the screen. In Oracle Utilities Application Framework V2.x (**true**), duplicate HTML ids were permitted on screen elements. In Oracle Utilities Application Framework V4.x (**false**), unique ids are generated for screen elements by default. Setting this value to **true** will emulate Oracle Utilities Application Framework V2.x behavior. This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.x and wish to retain that behavior.
Ordinality	Optional
Valid Values	[true|**false**]
Source	Manually set. Use user exit **spl.properties.exit.include** to set this value to **true**, if desired
Applicability	WEB
Example:	`spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs=false`

spl.runtime.disableCSRFProtection - Disable Security Tokens

Parameter	**spl.runtime.disableCSRFProtection**
Description	Enables or disables security token support for testing purposes with testing tools that do not support security tokens. It is highly recommended that this configuration setting be set to **false** in all environments, especially Production. It should only be set to **true** in testing environments where test tools are used and the test tool used does not support security tokens.
Ordinality	Optional
Valid Values	[true|**false**]
Source	Manually set. Use user exit **spl.properties.exit.include** to set this value to **true**, if desired
Applicability	WEB
Example:	`spl.runtime.disableCSRFProtection=false`

spl.runtime.environ.init.dir - Location of Configuration Files

Parameter	**spl.runtime.environ.init.dir**
Description	Location of the configuration files used for the product. This is used to allow the product to read configuration files at runtime to determine behavior.
Ordinality	Mandatory
Valid Values	URL is derived from **spl.runtime.environ.SPLEBASE** with the **etc** directory appended.
Source	Derived from **cistab** file.
Applicability	WEB BAS IWS
Example:	`spl.runtime.environ.init.dir=/spl/OUAFDEMO/etc`

spl.runtime.environ.isWebExpanded - Expanded Mode

Parameter	**spl.runtime.environ.isWebExpanded**
Description	Determines whether WAR/EAR files are built [**false**] or as expanded directories [**true**]. The expanded mode is used for development purposes only and is only supported on Oracle WebLogic. It is recommended to set this to **false** to use WAR/EAR files for deployment.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **WEB_ISEXPANDED** parameter from **ENVIRON.INI**.
Applicability	WEB BAS
Example:	`spl.runtime.environ.isWebExpanded=false`

spl.runtime.environ.SPLEBASE - Location Of Software

Parameter	**spl.runtime.environ.SPLEBASE**
Description	Home directory of the product installation
Ordinality	Mandatory
Valid Values	Full expanded directory on machine
Source	Derived from **cistab** file.
Applicability	WEB BAS IWS XAI BATCH
Example:	`spl.runtime.environ.SPLEBASE=/spl/OUAFDEMO`

spl.runtime.environ.SPLOUTPUT - Location Of Output

Parameter	**spl.runtime.environ.SPLOUTPUT**
Description	Home directory of the product output
Ordinality	Mandatory
Valid Values	Full expanded directory on machine
Source	Derived from **cistab** file.
Applicability	WEB BAS IWS XAI BATCH
Example:	spl.runtime.environ.SPLEBASE=/spl/splapp/OUAFDEMO

spl.runtime.management.batch.cluster.url - Batch Cluster URL

Parameter	**spl.runtime.management.batch.cluster.url**
Description	URL for the Batch Cluster
Ordinality	Mandatory
Valid Values	Valid URL for Batch Cluster
Source	Derived from **WEB_BATCH_CLUSTER_URL** parameter from **ENVIRON.INI**.
Applicability	WEB
Example:	spl.runtime.management.batch.cluster.url=

spl.runtime.management.connector.url.default - JMX URL

Parameter	**spl.runtime.management.connector.url.default**
Description	JMX formatted URL to be used for JMX based monitoring.
Ordinality	Optional
Valid Values	URL is generated from the following settings: Web Application Server **WEB_WLHOST** and **WEB_JMX_RMI_PORT_PERFORMANCE** Business Application Server **BSN_WLHOST** and **BSN_JMX_RMI_PORT_PERFORMANCE** Inbound Web Services **BSN_WLHOST** and **BSN_JMX_RMI_PORT_PERFORMANCE**
Source	Derived from **ENVIRON.INI** file.
Applicability	WEB BAS IWS
Example:	spl.runtime.environ.SPLEBASE= service:jmx:rmi:///jndi/rmi://localhost: 1100/oracle/ouaf/ejbAppConnector

spl.runtime.management.rmi.port - JMX Port

Parameter	**spl.runtime.management.rmi.port**
Description	JMX formatted URL to be used for JMX based monitoring.
Ordinality	Optional
Valid Values	URL is generated from the following settings: Web Application Server **WEB_WLHOST** Business Application Server **BSN_WLHOST** Inbound Web Services **BSN_WLHOST**
Source	Derived from **ENVIRON.INI** file.
Applicability	WEB BAS IWS
Example:	`spl.runtime.management.rmi.port=1100`

spl.runtime.options.isFCFEnabled - RAC FCF enabled

Parameter	**spl.runtime.options.isFCFEnabled**
Description	Enable or Disable RAC Fast Connection Failover support for database connections. Oracle Notification Services configuration is set by the **spl.runtime.options.onsserver=nodes** parameter.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **ONSCONFIG** parameter from **ENVIRON.INI**.
Applicability	BAS BATCH
Example:	`spl.runtime.options.isFCFEnabled=false`

spl.runtime.options.onsserver=nodes - ONS Configuration

Parameter	**spl.runtime.options.onsserver=nodes**
Description	Specify the Oracle Notification Services (ONS) configuration for Fast Connection Failover support for database connections. Refer to the Oracle Notification Services documentation for details of ONS setup and values for this setting.
Ordinality	Mandatory
Valid Values	Valid ONS configuration string (without the **nodes=** prefix)
Source	Derived from **ONSCONFIG** parameter from **ENVIRON.INI**.

Applicability	BAS BATCH
Example:	`spl.runtime.options.onsserver=nodes=racnode1:4200,racnode2:4200\nwalletfile= /oracle11/onswalletfile`

spl.runtime.options.allowSystemDateOverride - Set Testing Date

Parameter	`spl.runtime.options.allowSystemDateOverride`
Description	Allows the user of system test date override feature for testing. Refer to the Setting the Date for testing purposes for more information. It is highly recommended that this setting should be set to `false` in Production environments.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Manually set. Use custom templates to set this value to true, if desired.
Applicability	WEB BAS
Example:	`spl.runtime.options.allowSystemDateOverride=false`

spl.runtime.options.createSimpleWebAppContextEnable - Web Cache Loading

Parameter	`spl.runtime.options.createSimpleWebAppContextEnable`
Description	Enables or disables cache loading from Web Application Server. Setting this value to `true` will load the server cache through the Business Application Server. If the cache loading, is desired to be loaded within each tier then set this value to `false`. It is highly recommended that this setting should be set to `true` in Production environments.
Ordinality	Optional
Valid Values	[**true**|false]
Source	Manually set. Use user exit `spl.properties.exit.include` to set this value to `false`, if desired.
Applicability	WEB
Example:	`spl.runtime.options.createSimpleWebAppContextEnable=false`

spl.runtime.options.isDevelopmentMode - Development Mode

Parameter	`spl.runtime.options.isDevelopmentMode`
Description	Enables or disables Development Mode for the product. Development mode disables and enables key features within the architecture to allow the Oracle Utilities SDK to be used optimally. The following settings are affected by this parameter:

- Caching is disabled at all layers of the architecture. This can be renabled manually.
- Screen Preloading is disabled. This can be renabled manually.
- Compression is disabled at all layers of the architecture. This can be renabled manually.
- The product is set to expanded mode to allow developers access to individual J2EE objects rather than using WAR/EAR files.

It is highly recommended that this setting should be set to **false** in Production environments.

This setting is only set to **true** where the Oracle Utilities SDK is used directly.

Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Derived from **WEB_ISDEVELOPMENT** parameter from **ENVIRON.INI**.
Applicability	WEB BAS IWS XAI
Example:	`spl.runtime.options.isDevelopmentMode=false`

spl.runtime.oracle.statementCacheSize - Java SQL Cache Size

Parameter	**spl.runtime.oracle.statementCacheSize**
Description	Number of SQL statement cached for Java based code. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	1 - 2147483647 (Default: **300**)
Source	Derived from templates. Use custom templates to set this value to an alternative, if desired
Applicability	WEB BAS BATCH
Example:	`spl.runtime.oracle.statementCacheSize=300`

spl.runtime.performSignedNumberValidation.<product> - Number Validation

Parameter	**spl.runtime.performSignedNumberValidation.*<product>***
Description	Unsigned Number Validation routine used by product. By default a F1 based routine is provided and this setting allows products to provide their own processing. This setting is provided for products only. Implementations should not set this value.

Ordinality	Optional
Valid Values	[true|**false**]
Source	Derived from templates. Use user exits **spl.properties.service.exit.include** and **spl.properties.standalone.exit.include** to set this value to an alternative, if desired.
Applicability	BAS BATCH
Example:	`spl.runtime.performSignedNumberValidation.F1=true`

spl.runtime.service.extraInstallationServices - Installation Service

Parameter	**spl.runtime.service.extraInstallationServices**
Description	Specifies the internal service used for checking the installation records at startup time. This setting is used by the products, it is not recommended to be altered, unless otherwise directed by Oracle Support.
Ordinality	Optional
Valid Values	Service Name
Source	By default the service **CILTINCP** is used internally
Applicability	WEB
Example:	`spl.runtime.service.extraInstallationServices=CILTINCP`

spl.runtime.socket.file.dir - COBOL Socket Directory

Parameter	**spl.runtime.environ.file.dir**
Description	Location of the runtime directories. This setting is only used for products that support COBOL runtime.
Ordinality	Mandatory
Valid Values	URL is derived from **spl.runtime.environ.SPLEBASE** with the **runtime** directory appended.
Source	Derived from **cistab** file.
Applicability	WEB BAS
Example:	`spl.runtime.environ.init.runtime=/spl/OUAFDEMO/runtime`

spl.runtime.sql.highValue - SQL High Value

Parameter	**spl.runtime.sql.highValue**
Description	The high value used for paremeters by SQL queries.

	It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	Valid high value generated by installer.
Source	Derived from the **HIGHVALUE** setting in the **ENVIRON.INI**
Applicability	WEB BAS BATCH
Example:	`spl.runtime.sql.highValue=\uF8FF`

spl.runtime.utf8Database - UTF8 Support

Parameter	**spl.runtime.utf8Database**
Description	Enables or disables UTF8 support within the product.
Ordinality	Mandatory
Valid Values	[**true**|false]
Source	Derived from the **ENCODING** setting in the **ENVIRON.INI**
Applicability	WEB BAS IWS XAI BATCH
Example:	`spl.runtime.utf8Database=true`

spl.serviceBean.jndi.name - JNDI Name for Business Application Server

Parameter	**spl.serviceBean.jndi.name**
Description	JNDI name for Business Application Server generated by installer. It is highly recommended not to alter this value unless otherwise advised by Oracle Support.
Ordinality	Mandatory
Valid Values	Generated by installer.
Source	Derived from the **WEB_CONTEXT_ROOT** setting in the **ENVIRON.INI**
Applicability	WEB IWS XAI
Example:	`spl.serviceBean.jndi.name=ouaf/servicebean`

spl.tools.loaded.applications - Loaded product

Parameter	**spl.tools.loaded.applications**
Description	List the internal product codes installed under the Oracle Utilities Application Framework. This setting is maintained by the individual products and should not be altered unless otherwise instructed by Oracle Support.

Ordinality	Mandatory
Valid Values	Comma separated list of products
Source	Maintained by product templates
Applicability	WEB BAS IWS XAI BATCH
Example:	`spl.tools.loaded.applications=base,cm`

hibernate.properties

The hibernate.properties file defines the connection to the database and the Object to Relational database mapping.

hibernate.cache.use_second_level_cache - Use Cache

Parameter	`hibernate.cache.use_second_level_cache`
Description	Enable or Disable the second level cache, which is enabled by default for classes which specifies a cache mapping. This is disabled by default as Oracle Utilities Application Framework includes its own cache.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.cache.use_second_level_cache = false`

hibernate.connection.datasource - JNDI Data Source

Parameter	`hibernate.connection.datasource`
Description	Enables Data Source support and indicates JNDI data source. This is an alternative to UCP. See JNDI Data Sources for more information. JNDI data sources cannot be used for batch processing.
Ordinality	Mandatory
Valid Values	JNDI path to data source
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI
Example:	`hibernate.connection.datasource=jdbc/ouafdb`

hibernate.connection.driver_class - Connection Driver Class

Parameter	`hibernate.connection.driver_class`
Description	Driver class used by Hibernate.
Ordinality	Mandatory
Valid Values	Valid Driver Class. Default: **oracle.jdbc.driver.OracleDriver**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.driver_class = oracle.jdbc.driver.OracleDriver`

hibernate.connection.password - Database User Password

Parameter	`hibernate.connection.password`
Description	Encrypted password for Database User.
Ordinality	Mandatory
Valid Values	Generated Database Password. Value depends on channel: Online uses **DBPASS** XAI/IWS uses **XAI_DBPASS** Batch uses **BATCH_DBPASS**
Source	Maintained by product templates. Set appropriate value for user required in required parameter.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.password = ENCKS(….)`

hibernate.connection.provider_class - Connection Provider

Parameter	`hibernate.connection.provider_class`
Description	Connection Provider Class for Hibernate. Use the default UCP provider as supplied unless using JBDC Data Sources where `com.hibernate.connection.DatasourceConnectionProvider` is used instead.
Ordinality	Mandatory
Valid Values	Generated Provider Class. Default: **com.splwg.shared.common.UCPConnectionProvider**
Source	Maintained by product templates. Use custom templates to implement a different value.

Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.provider_class=` `com.splwg.shared.common.UCPConnectionProvider`

hibernate.connection.release_mode - Connection Release

Parameter	`hibernate.connection.release_mode`
Description	This parameter controls when a connection is released to the pool. By default the value is set to **after_transaction**. Whilst other values can be used, for consistency of the connection it is recommended to use the default.
Ordinality	Mandatory
Valid Values	Generated Release Mode. Default: **after_transaction**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.release_mode=after_transaction`

hibernate.connection.url - JDBC URL

Parameter	`hibernate.connection.url`
Description	JDBC URL for connection to database.
Ordinality	Mandatory
Valid Values	Generated URL or using override database connection.
Source	Maintained by product templates. Use **DB_OVERRIDE_CONNECTION** for specific values.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.url =` `jdbc:oracle:thin:@ouafdemo.us.oracle.com:1522:/MYPDB`

hibernate.connection.username - Database User

Parameter	`hibernate.connection.username`
Description	JDBC User for connection.
Ordinality	Mandatory
Valid Values	The value depends on the channel accesssed: Online uses **DBUSER** XAI/IWS uses **XAI_DBUSER** Batch uses **BATCH_DBUSER**

Source	Maintained by product templates. Set appropriate value for user required in required parameter.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.connection.username = CISUSER`

hibernate.dialect - Dialect Used

Parameter	`hibernate.dialect`
Description	This is the SQL dialect (database type) for the database being used. Any valid Hibernate dialect may be used. Refer to http://www.hibernate.org/hib_docs/v3/api/org/hibernate/dialect/package-summary.html for a full list.
Ordinality	Mandatory
Valid Values	Generated Dialect. Default: **org.hibernate.dialect.Oracle10gDialect**.
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.dialect = org.hibernate.dialect.Oracle10gDialect`

hibernate.jdbc.batch_size - JDBC2 Update Support

Parameter	`hibernate.jdbc.batch_size`
Description	A non-zero value enables use of JDBC2 batch updates by Hibernate.
Ordinality	Mandatory
Valid Values	Generated Value. Default: **30**.
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.jdbc.batch_size = 30`

hibernate.jdbc.fetch_size - Solution Set Size

Parameter	`hibernate.jdbc.fetch_size`
Description	Determines a hint to the JDBC driver on the the number of rows to return in any SQL statement. This is overridden at runtime on zone parameters and commit interval.
Ordinality	Mandatory
Valid Values	Generated Value. Default: **100**.

Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.jdbc.fetch_size = 100`

hibernate.max_fetch_depth - Join Fetch Depth

Parameter	`hibernate.max_fetch_depth`
Description	Sets a maximum *depth* for the outer join fetch tree for single-ended associations (one-to-one, many-to-one). A 0 disables default outer join fetching
Ordinality	Mandatory
Valid Values	Generated Value. Default: **2**.
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.max_fetch_depth = 2`

hibernate.query.factory_class - HQL Parser

Parameter	`hibernate.query.factory_class`
Description	The classname of a Querey Factory to use for HQL parsing.
Ordinality	Mandatory
Valid Values	Generated Value. Default: **org.hibernate.hql.internal.classic.ClassicQueryTranslatorFactory**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.query.factory_class= org.hibernate.hql.internal.classic.ClassicQueryTranslatorFactory`

hibernate.query.substitutions - Literal Mappings

Parameter	`hibernate.query.substitutions`
Description	Mapping from tokens in Hibernate queries to SQL tokens (tokens might be function or literal names, for example).
Ordinality	Mandatory
Valid Values	Generated Value. Default: **true 'Y', false 'N'**
Source	Maintained by product templates. Use custom templates to implement a

	different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.query.substitutions = true 'Y', false 'N'`

hibernate.show_sql - Debug Mode

Parameter	`hibernate.show_sql`
Description	Use of Debug Mode for all SQL Statements. It is highly recommended not to enable debug at this level.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.show_sql = false`

hibernate.transaction.factory_class - Transaction Class

Parameter	`hibernate.transaction.factory_class`
Description	The classname of a Transaction Factory to use with Hibernate Transaction API.
Ordinality	Mandatory
Valid Values	Generated Value. Default: **org.hibernate.transaction.JDBCTransactionFactory**.
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.transaction.factory_class = org.hibernate.transaction.JDBCTransactionFactory`

hibernate.ucp.connection_wait_timeout - UCP Connection Timeout

Parameter	`hibernate.ucp.connection_wait_timeout`
Description	Specifies how long, in seconds, an application request waits to obtain a connection if there are no longer any connections in the pool
Ordinality	Mandatory
Valid Values	Generated Value. Default: **5**.
Source	Maintained by product templates. Use custom templates to implement a different value.

Applicability	BAS IWS XAI BATCH
Example:	`hibernate.ucp.connection_wait_timeout=5`

hibernate.ucp.inactive_connection_timeout - Connection Inactivity Timeout

Parameter	`hibernate.ucp.inactive_connection_timeout`
Description	Specifies how long, in seconds, an available connection can remain idle before it is closed and removed from the pool.
Ordinality	Mandatory
Valid Values	Generated Value. Default: **300**.
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.ucp.inactive_connection_timeout=300`

hibernate.ucp.jmx_enabled - Enable UCP JMX

Parameter	`hibernate.ucp.jmx_enabled`
Description	Enable or Disable JMX Mbeans for UCP. The Mbeans will appear as additional classes on the existing JMX capabilities for that tier.
Ordinality	Mandatory
Valid Values	[true|**false**]
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.ucp.jmx_enabled=false`

hibernate.ucp.max_size - Maximum Pool Size

Parameter	`hibernate.ucp.max_size`
Description	Maximum Pool size for this teir. Maximum number of active connections to the database.
Ordinality	Mandatory
Valid Values	Generated. Default: **30**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	`hibernate.ucp.max_size=30`

hibernate.ucp.max_statements - SQL Buffer size

Parameter	hibernate.ucp.max_statements
Description	Number of statements cached for performance on this teir.
Ordinality	Mandatory
Valid Values	Generated. Default: **50**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	hibernate.ucp.max_statements=50

hibernate.ucp.min_size - Minimum Pool Size

Parameter	hibernate.ucp.min_size
Description	Initial Connection Pool size for this teir.
Ordinality	Mandatory
Valid Values	Generated. Default: **1**
Source	Maintained by product templates. Use custom templates to implement a different value.
Applicability	BAS IWS XAI BATCH
Example:	hibernate.ucp.min_size=1