

Server Administration Guide

Oracle Utilities Market Settlements Management
Version 2.3.0.2 (OUAF 4.4.0.2.0)

F21780-01

September 2019

ORACLE[®]

Server Administration Guide, Oracle Utilities Market Settlements Management, Version 2.3.0.2 (OUAF 4.4.0.2.0)

F21780-01

Copyright © 2007-2019 Oracle. All rights reserved.

Primary Author: Oracle Utilities Global Business Unit

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for:

(a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents

| | |
|---|-----------|
| Preface | 15 |
| Introduction | 15 |
| Updates to This Documentation | 15 |
| Other Documentation | 15 |
| Whitepapers | 16 |
| Architecture | 18 |
| Roles and Features | 20 |
| Client..... | 20 |
| Web Application Server | 20 |
| Business Application Server | 21 |
| Mobile Application..... | 21 |
| Database Server..... | 22 |
| Batch Server BATCH | 22 |
| Concepts | 26 |
| Environment | 26 |
| Environment Identifier | 26 |
| Server Roles..... | 26 |
| Administration User Id and Group | 27 |
| Native Support | 28 |
| Directory Structure | 28 |
| Software Directory Structure | 29 |
| Software (splapp) Directory..... | 31 |
| Tools directory..... | 33 |
| Directory Permissions | 33 |
| Output Structure | 34 |
| Environment Variables | 35 |
| Common Application Logs | 37 |
| Batch Concepts BATCH | 38 |
| Batch Clustering Concepts..... | 39 |
| Batch Controls..... | 40 |
| Standard parameters..... | 40 |
| Additional Key Parameters | 41 |
| Batch Run Tree | 43 |
| Timed Jobs | 43 |
| Level Of Service..... | 43 |
| Explanation of Commit Interval..... | 44 |
| Explanation of Thread Limit and Thread Number | 44 |
| Explanation of Restart and Rerun | 45 |
| Submission Methods | 45 |
| Operations | 49 |
| Command Line Utilities | 49 |
| bedit - Batch Edit BATCH | 49 |
| buildJavadocsIndex[.sh] - Builds Appviewer java docs..... | 50 |
| configureEnv – Setup Environment settings..... | 51 |
| createDBStandalone[.sh] - Creates Database Patches..... | 51 |
| extractCMSourceUNIX[.sh] - Migrate Code Across Upgrades | 52 |
| genappvieweritems – generate AppViewer | 53 |
| initialSetup – Maintain Configuration Settings | 54 |

| | |
|---|------------|
| installSFGGroup[.sh] - Install Software Patches | 56 |
| invokeDBUpdatePatch[.sh] - Regenerates passwords in Installation Options | 57 |
| iwsdeploy - Inbound Web Services Deployment | 57 |
| jmxbatchclient[.sh] – JMX batch command line BATCH | 58 |
| ouafDatabasePatch[.sh] - Install Database Patches | 59 |
| PatchDeployUtility[.sh] - Create and Deploy Patch Rollups | 60 |
| showjavaproc[.sh] - Show java processes | 61 |
| splenviron – Set Environment variables | 62 |
| starttpw[.sh] - Start Threadpoolworker | 64 |
| submitjob[.sh] - Submit Batch Threads BATCH | 64 |
| threadpoolworker[.sh] - Manage Threadpools BATCH | 68 |
| Starting and Stopping Using consoles | 70 |
| Monitoring | 71 |
| Monitoring Regimes..... | 71 |
| Monitoring Client Machines..... | 72 |
| Monitoring The Desktop..... | 72 |
| Client Debug facility..... | 73 |
| Monitoring Web/business Application Server | 75 |
| JMX Based Monitoring | 75 |
| Web Application Server JMX Reference | 77 |
| Business Application Server JMX Reference | 86 |
| Batch JMX Reference BATCH | 89 |
| JMX Security..... | 95 |
| Extending JMX Security..... | 96 |
| Execution Dump Format | 96 |
| Service Lists | 97 |
| Resetting Statistics | 98 |
| Database Connection Monitoring..... | 98 |
| Probe Transaction..... | 99 |
| Configuration | 100 |
| Configuration Files | 100 |
| cistab - Global Configuration Files | 100 |
| coherence-cache-config.xml - Oracle Coherence Configuration BATCH | 101 |
| e0Batch.properties - General Batch Properties BATCH | 101 |
| ejb-jar.xml - Enterprise Java Server Beans Configuration..... | 101 |
| ENVIRON.INI - Environment Configuration File..... | 101 |
| hibernate.properties - Database Connectivity Settings | 103 |
| log4j2.properties – Product Logging Configuration..... | 103 |
| Server Jar File (ouaf_jar_versions.txt)..... | 103 |
| spl.properties – Product Behaviour Settings..... | 104 |
| submitbatch.properties - Submitter Configuration BATCH | 104 |
| tangosol-coherence-override.xml - Coherence Overrides BATCH | 104 |
| threadpoolworker.properties - Threadpool Configuration BATCH | 104 |
| web.xml – JEE Deployment Descriptor | 105 |
| weblogic.xml – Oracle WebLogic Extensions..... | 106 |
| weblogic-ejb-jar.xml - WebLogic extensions for Enterprise Java Server Beans | 108 |
| webservices.xml.resource - Web Services configuration | 108 |
| Web Browser Configuration..... | 108 |
| Web Application Server Configuration..... | 111 |
| Caveat | 111 |
| Web Application Server Concepts..... | 112 |
| Web Applications | 112 |

| | |
|---|------------|
| Web Application Server Configuration Files | 112 |
| Quick Reference Guide for Web Application Server Configuration | 116 |
| User Interface Backward Compatibility Settings | 117 |
| Web Application Server Deployment Process | 117 |
| Business Application Server Configuration | 118 |
| Business Application Server Concepts | 120 |
| Business Application Server Configuration Files | 120 |
| Business Application Server Configuration Process | 120 |
| Quick Reference Guide for Business Application Server Configuration | 123 |
| Business Application Server Deployment Process | 123 |
| Inbound Web Services Configuration | 124 |
| Enabling Inbound Web Services | 125 |
| Inbound Web Services Configuration Files | 126 |
| Inbound Web Services Concepts | 127 |
| Inbound Web Services Configuration Process | 128 |
| Inbound Web Services Deployment Process | 131 |
| Quick Reference Guide for Inbound Web Services Configuration | 132 |
| Inbound Web Services Deployment Utilities | 132 |
| Batch Server Configuration BATCH | 133 |
| Batch Configuration Files | 133 |
| General Configuration Process | 134 |
| Configuring Batch using bedit | 135 |
| Batch Configuration Guidelines | 142 |
| Oracle Cloud Support | 144 |
| Support for host whitelisting | 144 |
| Support for URI specification using Substitution Variables | 146 |
| Defining File Alias Using Extendable Lookup | 148 |
| Support for Oracle Cloud Storage Access | 149 |
| Consolidated logging | 149 |
| Domain Template Support | 149 |
| Diagnostics Support (WLDF) | 150 |
| ECID Support | 151 |
| Global Cache Flush Support | 152 |
| Java Flight Recorder Support | 153 |
| Work Manager Support | 153 |
| Trust Store Support | 154 |
| Miscellaneous Operations And Configuration | 155 |
| Enabling Email Logging from Log4j2 | 155 |
| Overriding the default Oracle database connection information | 156 |
| Centralized Properties Customization | 157 |
| Cache Management | 158 |
| Server Cache | 158 |
| Client Cache | 160 |
| Oracle WebLogic: Expanded or Archive Format | 161 |
| Implementing Custom Templates | 162 |
| Additional templates | 163 |
| Sending emails at the conclusion of batch process BATCH | 165 |
| Oracle WebLogic Configuration Support | 167 |
| Using Configuration Files outside the WAR/EAR file | 168 |
| Oracle RAC Support | 169 |
| Adding a custom Privacy policy screen | 170 |
| User Exit Include Files | 171 |
| Properties File User Exits | 174 |
| Custom JMS configuration | 175 |
| Online Transaction Timeouts | 177 |

| | |
|--|------------|
| Setting the Date for testing purposes | 178 |
| Secure Transactions | 180 |
| Native Oracle WebLogic Support | 180 |
| Manual Re-Deployment Process..... | 182 |
| Oracle WebLogic Console Operations | 182 |
| Starting and Stopping from console..... | 182 |
| Monitoring Web Applications using the console..... | 184 |
| Specifying custom log file names BATCH | 184 |
| Turning off L2 Cache BATCH | 185 |
| JDBC Support | 185 |
| Cancelling Batch Processes Using JMX BATCH | 187 |
| Operations from Oracle Enterprise Manager | 188 |
| Discovering Oracle Utilities targets | 188 |
| Discovering Oracle WebLogic Targets | 189 |
| Oracle Utilities Targets | 190 |
| Monitoring using Oracle Enterprise Manager | 190 |
| Oracle Scheduler Integration | 192 |
| Introduction | 192 |
| Background Materials | 192 |
| Oracle Scheduler Interface Overview | 192 |
| Oracle Scheduler Configuration..... | 194 |
| Oracle Scheduler Execution..... | 195 |
| Oracle Scheduler Interface (OUAF_BATCH) | 195 |
| Submit_Job - Submit a Job or Thread..... | 196 |
| Restart_Thread - Restart a failed thread/job | 198 |
| Cancel_Job - Cancel a running Job..... | 199 |
| Set_Option - Set a global or Job option..... | 199 |
| Unset_Option - Reset a global or Job option..... | 201 |
| Raise_Thread_Error - Raise an error | 201 |
| Setting Preferences | 201 |
| Order Of Preference | 202 |
| Using soft_parm_map | 202 |
| Using soft_parm_list..... | 203 |
| Oracle Scheduler – OUAF REST APIs | 203 |
| Building a Schedule | 211 |
| Recommended tools..... | 211 |
| Defining Programs | 212 |
| Defining Job Classes..... | 213 |
| Defining Schedules..... | 214 |
| Defining Remote Destinations and Credentials..... | 215 |
| Defining Jobs..... | 217 |
| Defining Chains | 218 |
| Other Scheduler Objects | 219 |
| Executing Jobs | 220 |
| Monitoring Oracle Scheduler | 220 |
| Job States | 220 |
| Dictionary Views..... | 220 |
| Scheduler Notifications | 221 |
| Cancellation | 221 |
| Exporting Schedules across environments | 221 |
| Appendices - Parameters | 222 |
| Conventions in this section | 222 |
| ENVIRON.INI | 222 |

| | |
|---|-----|
| ALLOW_SELFSIGNED_SSL – enable self signed SSL | 222 |
| ANT_ADDITIONAL_OPT - Additional options for ANT | 222 |
| ANT_HOME - Location of ANT | 223 |
| ANT_OPT_MAX - ANT Maximum Heap Size | 223 |
| ANT_OPT_MIN - ANT Minimum Heap Size | 223 |
| appViewer - Name of appViewer WAR file | 223 |
| BATCHDAEMON - Whether Online Batch Daemon is enabled | 224 |
| BATCHEDIT_ENABLED - Whether Batch Edit Facility is Enabled | 224 |
| BATCHENABLED - Whether Online Batch Server is Enabled | 224 |
| BATCHTHREADS - Number of Online Batch Server Threads | 224 |
| BATCH_DBPASS - Batch User Password | 224 |
| BATCH_DBUSER - Batch User | 225 |
| BATCH_MEMORY_ADDITIONAL_OPT - Threadpool Worker JVM additional options | 225 |
| BATCH_MEMORY_OPT_MAX - Threadpool Worker Java Maximum Heap Size | 225 |
| BATCH_MEMORY_OPT_MIN - Threadpool Worker Java Minimum Heap Size | 225 |
| BATCH_MODE - Default Mode of Batch | 226 |
| BATCH_ONLINE_LOG_DIR - Online Batch Submission Log Directory | 226 |
| BATCH_RMI_PORT - RMI Port for Batch | 226 |
| BSN_APP - Business Server Application Name | 226 |
| BSN_JMX_RMI_PORT_PERFORMANCE - JMX Port for Business Application Server | 226 |
| BSN_JMX_SYSPASS - Password for BAS JMX Monitoring | 227 |
| BSN_JMX_SYSUSER - Default User for BSN JMX Monitoring | 227 |
| BSN_WLHOST - Business App Server Host | 227 |
| CALC_WLS_COMMON_JAR_DIR – Weblogic’s Jersey Jars directory | 227 |
| CALC_WLS_SERVER_JAR_DIR – Weblogic server lib jars directory | 228 |
| CHAR_BASED_DB – specifies if the DB is char based | 228 |
| CLOUD_CUSTOM_WHITE_LIST_PATH - Custom URL Whitelists for Cloud | 228 |
| CLOUD_LOCATION_F1_BASE_IWS_URL – URI for variable F1_BASE_IWS_URI | 229 |
| CLOUD_LOCATION_F1_BASE_REST_URL – URI for variable F1_BASE_REST_URL | 229 |
| CLOUD_LOCATION_F1_BASE_WEB_URI – URI for variable F1_BASE_WEB_URI | 229 |
| CLOUD_LOCATION_F1_MIGR_ASSISTANT_FILES – Location of CMA files for migrations | 229 |
| CLOUD_LOCATION_F1_OAUTH2_URI – URI for OAUTH2 | 230 |
| CLOUD_RESTRICTIONS_URLS_ENABLE - Enable or Disable URL Restrictions | 230 |
| CLOUD_SUBSTITUTION_VARIABLE_LIST_FILE_LOCATION - Enable or Disable URL Restrictions | 230 |
| CLOUD_WHITE_LIST_PATH - URL Whitelists for Cloud | 231 |
| CMPDB - Database Type | 231 |
| CM_INSTALL_SAMPLE - Whether samples are installed | 231 |
| COHERENCE_CLUSTER_ADDRESS - Multicast address for Batch Cluster | 231 |
| COHERENCE_CLUSTER_MODE - Batch Cluster Mode | 232 |
| COHERENCE_CLUSTER_NAME - Batch Cluster Name | 232 |
| COHERENCE_CLUSTER_PORT - Batch Cluster Port Number | 232 |
| CONTEXTFACTORY - Context Factory for Mobile Application | 233 |
| CONSOLIDATED_LOG_FILE_PATH- Directory for consolidated log file | 233 |
| CSRF_PROTECTION – Cross-Site Request Forgery protection | 233 |
| CUSTOM_SQL_SECURITY – Enable custom SQL security | 233 |
| DBCONNECTION - JDBC Connection string | 234 |
| DBDRIVER - Database Driver | 234 |
| DBNAME - Database Name | 234 |
| DBPASS - Online Database User Password | 234 |
| DBPASS_GEOCODE_WLS - GEOCODE Database Password | 235 |
| DBPASS_MDS - MDS Database Password | 235 |
| DBPASS_ORASDPM - DPM Database Password | 235 |
| DBPASS_OSB - OSB Database Password | 236 |
| DBPASS_SOAINFRA - SOA Database Password | 236 |
| DBPORT - Database Listener Port | 236 |
| DBSERVER - Database Server | 237 |
| DBURL_GEOCODE - GEOCODE JDBC URL | 237 |

| | |
|--|-----|
| DBURL_OSB - OSB JDBC URL | 237 |
| DBURL_SOA - SOA Suite JDBC URL | 237 |
| DBUSER - Online Database User | 238 |
| DBUSER_GEOCODE - GEOCODE Database User | 238 |
| DBUSER_MDS - SOA MDS Database User | 238 |
| DBUSER_ORASDPM - DPM Database User | 238 |
| DBUSER_OSB - OSB Database User | 239 |
| DBUSER_SOAINFRA - SOA Database User | 239 |
| DB_OVERRIDE_CONNECTION - Custom JDBC URL | 239 |
| DESC - Environment Description | 240 |
| DIALECT - Hibernate Dialect | 240 |
| DIRSEP - Directory separator | 240 |
| DOC1BILLSCRIPT - Name Of External Print Script | 241 |
| DOC1SCRIPT - Name Of Generic Print Script | 241 |
| ENABLE_JMS_GLOBAL_FLUSH – Enable Flusing of JMS | 241 |
| ENCODING - Whether Encryption Is Enabled | 241 |
| ENVIRONMENT_ID - Environment Identifier | 242 |
| FORCE_HTTP – Enforce http protocol | 242 |
| FW_VERSION - Oracle Utilities Application Framework version | 242 |
| FW_VERSION_NUM - Oracle Utilities Application Framework Version Identifier | 243 |
| GIS - GIS Support | 243 |
| GIS_URL - GIS Service URL | 243 |
| GIS_WLSYSPASS - GIS WebLogic System Password | 243 |
| GIS_WLSYSUSER - GIS WebLogic System User Id | 244 |
| GLOBAL_JVMARGS - Additional JVM Arguments | 244 |
| HEADEND_CD_CB - Headend System URI for SOA Configuration Plan (CD_CB) | 244 |
| HEADEND_MR_CB - Headend System URI for SOA Configuration Plan (MR_CB) | 245 |
| HIBERNATE_JAR_DIR - Location of Hibernate JAR files | 245 |
| HIGHVALUE - Language specific highvalues | 245 |
| HSTS_MAX_AGE – Maximum age of strict transport security | 245 |
| HSTS_PRELOAD – Strict transport security preload | 246 |
| HSTS_SUBDOMAINS – Subdomains included in strict transport security | 246 |
| IWS_JDBC_NAME - JNDI Name for the IWS Connection pools | 246 |
| IPCSTARTPORT - Starting IPC Port for Scheduler | 246 |
| IWSWAR - Name of Inbound Web Services WAR | 247 |
| JAVAENCODING - Java Language Encoding | 247 |
| JAVA_HOME - Location of Java SDK | 247 |
| JAVA_VENDOR - Java Vendor | 247 |
| JAVA_VERSION - JVM Version string | 248 |
| JAVA_VERSION_MAIN - Abberviated JVM Version | 248 |
| JDBC_NAME - JNDI Name for the Online JDBC Data Source | 248 |
| JDBC_TIMEOUT – JDBC read timeout | 248 |
| JNDI_GEOCODE - Geocode Data Source JNDI | 249 |
| JNDI_OSB - OSB Data Source JNDI | 249 |
| KS_ALIAS - Key Alias | 249 |
| KS_ALIAS_KEYALG - Key Algorithm | 249 |
| KS_ALIAS_KEYSIZE - Keysize | 250 |
| KS_HMAC_ALIAS - HMAC Key Alias | 250 |
| KS_HMAC_ALIAS_KEYALG - HMAC Key Algorithm | 250 |
| KS_HMAC_ALIAS_KEYSIZE - HMAC Keysize | 250 |
| KS_IMPORT_KEYSTORE_FOLDER - Keystore Import Folder | 251 |
| KS_KEYSTORE_FILE - Keystore File Name | 251 |
| KS_MODE - Key Padding Mode | 251 |
| KS_PADDING - Key Padding Algorithm | 251 |
| KS_STOREPASS_FILE - Key Password Store Pass | 252 |
| KS_STORETYPE - Key Store Type | 252 |
| LD_LIBRARY_PATH - Library Path for Windows/Linux/Solaris | 252 |

| | |
|--|-----|
| MAPDIR - Location of Map files used for Scheduler | 252 |
| MAPVIEWER_EAR - Location of Mapviewer EAR file..... | 253 |
| MAPVIEWER_ISLOCAL - Deploy Mapviewer Locally..... | 253 |
| MAXPROCESSINGTIME - Maximum Scheduler Processing Time | 253 |
| MINREQUESTS - Initial Scheduler Minimum Requests | 253 |
| MOBILE_APPDIR – Directory of the mobile application install | 254 |
| MOBILE_APP_ONLY - Deploy Only Mobility Web Application | 254 |
| MOBILE_ENABLED – If Mobile application is enabled | 254 |
| MODULES - Names of Modules installed (Deprecated)..... | 254 |
| NLS_LANG - NLS Language setting | 255 |
| NODEID - Scheduler Node Identifier | 255 |
| ONLINE_DISPLAY_HOME - Location of Print Rendering Software..... | 255 |
| ONS_JAR_DIR - Location of Oracle Notification Service Libraries | 255 |
| ONSCONFIG - ONS Configuration | 256 |
| OPSYS - Operating System..... | 256 |
| ORACLE_CLIENT_HOME - Oracle Client Home | 256 |
| OSB_HOME - Oracle Service Bus Home | 257 |
| OSB_HOST - Oracle Service Bus Host Name..... | 257 |
| OSB_LOG_DIR - Default Location For Oracle Service Bus Logs..... | 257 |
| OSB_PASS_WLS - Password for Oracle Service Bus User..... | 257 |
| OSB_PORT_NUMBER - Oracle Service Bus Port Number | 258 |
| OSB_USER - User Name for Oracle Service Bus..... | 258 |
| OUAF_DBMS_SCHEDULER_USER – DB User of DBMS scheduler..... | 258 |
| OWSM_ENABLED – Enable Oracle Web Services Manager | 259 |
| OWSM_PROTECTION_FOR_REST_SERVICES – Enable Oracle Web Services Manager protection for REST..... | 259 |
| PERLHOME - Location of PERL | 259 |
| SERVER_ROLES - Which server roles are enabled | 259 |
| SERVER_ROLE_BATCH - Whether the Batch Server is enabled in this installation | 260 |
| SERVER_ROLE_INTEGRATION - Whether the IWS Server is enabled in this installation | 260 |
| SERVER_ROLE_ONLINE - Whether the online server is enabled in this installation | 260 |
| SOA_HOME - Location Of Oracle SOA Suite | 260 |
| SOA_HOST - Hostname for Oracle SOA Suite..... | 260 |
| SOA_PORT_NUMBER - Port Number for Oracle SOA Suite..... | 261 |
| SPLADMIN - Administration Userid | 261 |
| SPLADMINGROUP - Administration Group | 261 |
| SPLApp - Name of Online WAR file..... | 261 |
| SPLDIR - Home Directory Of Product | 262 |
| SPLDIROUT - Location Of Product Output..... | 262 |
| SPLENVIRON - Environment Identifier | 262 |
| SPLOUTPUT - Directory for product output | 262 |
| SPLSERVICEAPP - Name of Business Application Server Application | 263 |
| SPLWEBAPP - Name of Web Application Server Application..... | 263 |
| STRICT_TRANSPORT_SECURITY – Enable Strict Transport Security | 263 |
| STRIP_HTML_COMMENTS - Strip Out Comments In Code | 263 |
| TEMPSTORAGE_SPLITFILESIZE - Split File Size in MB | 264 |
| TIMEOUT - Mobile Connection Timeout | 264 |
| TOP_VERSION - Product Version | 264 |
| TOP_VERSION_NUM - High level Product Version | 264 |
| URL - Mobile JNDI URL..... | 265 |
| WEB_ADMIN_SERVER – Name of the Web Admin Server | 265 |
| WEB_APP - Name of WebSphere Server Application..... | 265 |
| WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE - AppViewer Form Login Error Page..... | 265 |
| WEB_APPVIEWER_FORM_LOGIN_PAGE - AppViewer Form Login Page | 265 |
| WEB_APPVIEWER_PRINCIPAL_NAME - Application Viewer Principal Name | 266 |
| WEB_APPVIEWER_ROLE_NAME - Application Viewer Security Role..... | 266 |
| WEB_BATCH_CLUSTER_URL - Batch Cluster URL..... | 266 |

| | |
|--|------------|
| WEB_CONTEXT_ROOT - Web Context Root..... | 266 |
| WEB_DEPLOY_APPVIEWER - Deploy AppViewer or not | 267 |
| WEB_ENABLE_HEALTHCHECK - Enable System Healthcheck | 267 |
| WEB_FORM_LOGIN_ERROR_PAGE - Default Form Login Error Page | 267 |
| WEB_FORM_LOGIN_PAGE - Default Form Login Page..... | 267 |
| WEB_HELP_FORM_LOGIN_ERROR_PAGE - Help Form Login Error Page | 268 |
| WEB_HELP_FORM_LOGIN_PAGE - Help Form Login Page | 268 |
| WEB_ISAPPVIEWER - Enable AppViewer | 268 |
| WEB_ISDEVELOPMENT - Development Environment..... | 268 |
| WEB_IWS_MDB_RUNAS_USER - Default User for MDB to run as within product | 269 |
| WEB_IWS_SUPER_USERS - Users to run as in IWS | 269 |
| WEB_JMX_RMI_PORT_PERFORMANCE - JMX Port for Web Application Server monitoring | 269 |
| WEB_L2_CACHE_MODE - Turn off L2 Cache for Online Submission | 270 |
| WEB_MAXAGE - Text Cache | 270 |
| WEB_MAXAGEI - Image Cache | 270 |
| WEB_MOBILE_FORM_LOGIN_ERROR_PAGE - Default Mobile Form Login Error Page | 270 |
| WEB_MOBILE_FORM_LOGIN_PAGE - Default Form Login Page | 271 |
| WEB_PRELOADALL - Preload all pages On startup | 271 |
| WEB_PRINCIPAL_NAME - Default JEE Authorization Principal | 271 |
| WEB_ROLE_NAME - Default JEE Authorization Role | 271 |
| WEB_SERVER_HOME - Location Of Web Application Server | 271 |
| WEB_SERVER_VERSION - Web Server Version | 272 |
| WEB_SERVER_VERSION_MAIN - | 272 |
| WEB_WLAUTHMETHOD - Authentication Method..... | 272 |
| WEB_WLPAGECHECKSECONDS - Interval for recompilation of JSP | 273 |
| WEB_WLPORT - Web Server HTTP Port | 273 |
| WEB_WLSSLPORT - Oracle WebLogic SSL HTTP Port..... | 273 |
| WEB_WLSYSPASS - Oracle WebLogic JNDI System Password | 273 |
| WEB_WLSYSUSER - Oracle WebLogic JNDI System Userid | 274 |
| WLS_CLUSTER_NAME - Name of Cluster for Product | 274 |
| WLTHININT3CLIENT_JAR_DIR - Location of Thin WebLogic T3 Client..... | 274 |
| WL_HOME - Oracle WebLogic Home | 274 |
| XQUERIES_TO_CACHE - Size of XQuery Cache..... | 275 |
| XQUERY_CACHE_FLUSH_TIMEOUT - Cache Life for XQueries | 275 |
| Web.xml | 275 |
| Filter Mappings - JEE Filters | 275 |
| disableCompression - Disable Compression..... | 276 |
| maxAgeI - Max Age for Images | 277 |
| maxAge - Max Age for pages..... | 277 |
| auth-method - Authorization Method..... | 278 |
| fieldValuesAge - Server Cache Lifetime..... | 279 |
| disableUIPageCompression - Disable UI Compression | 279 |
| spl.properties | 279 |
| calendar.japanese.eras - Japanese Era Definition | 280 |
| com.oracle.ouaf.customer.whitelist.file – Customer's URL whitelist file | 280 |
| com.oracle.ouaf.consolidatedLog.fileName – Consolidate log file | 281 |
| com.oracle.ouaf.dbms.scheduler.schema – DBMS Scheduler object's schema | 281 |
| com.oracle.ouaf.file.tempstorage.dir – location for temporary file | 281 |
| com.oracle.ouaf.file.tempstorage.splitfilesize – split file size in MB | 282 |
| com.oracle.ouaf.migrationAssistant.exportDirectory – Directory for exported CMA file..... | 282 |
| com.oracle.ouaf.migrationAssistant.importDirectory – Directory for CMA file for import | 282 |
| com.oracle.ouaf.migrationAssistant.fileExtension – Extension of CMA file | 282 |
| com.oracle.ouaf.mobile.appdir – directory for mobile application..... | 283 |
| com.oracle.ouaf.proxy.password– Proxy credentials for senders if JVM is configured to use proxy..... | 283 |
| com.oracle.ouaf.proxy.user – Proxy credentials for senders if JVM is configured to use proxy..... | 283 |
| com.oracle.ouaf.system.database.disableTagging - Disables Database tagging..... | 284 |
| com.oracle.ouaf.system.keystore.alias - Keystore Alias..... | 284 |

| | |
|--|-----|
| com.oracle.ouaf.system.keystore.alias_keyalg - Keystore Alias Algorithm | 284 |
| com.oracle.ouaf.system.keystore.alias_keysize – Keystore Keysize Strength | 285 |
| com.oracle.ouaf.system.keystore.hmac_alias_keyalg - HMAC Algorithm | 285 |
| com.oracle.ouaf.system.keystore.hmac_alias_keysize - HMAC Key Size | 285 |
| com.oracle.ouaf.system.keystore.hmac_key_alias - HMAC alias | 286 |
| com.oracle.ouaf.system.keystore.file - Location Of Keystore | 286 |
| com.oracle.ouaf.system.keystore.mode - Keystore Mode..... | 286 |
| com.oracle.ouaf.system.keystore.padding - Key Padding..... | 287 |
| com.oracle.ouaf.system.keystore.passwordFileName - Password File | 287 |
| com.oracle.ouaf.system.keystore.type - Keystore Type | 287 |
| com.oracle.ouaf.system.truststore.alias - Truststore Alias | 288 |
| com.oracle.ouaf.system.truststore.alias_keyalg - Truststore Key Algorithm | 288 |
| com.oracle.ouaf.system.truststore.alias_keysize - Truststore Key Size..... | 288 |
| com.oracle.ouaf.system.truststore.file - Location Of Truststore | 289 |
| com.oracle.ouaf.system.truststore.passwordFileName - Location Of Truststore Password file..... | 289 |
| com.oracle.ouaf.truststore.type - Type Of Truststore..... | 289 |
| com.oracle.ouaf.ui.disableRequiredFieldIndicators – Disable required field indicators on UI map..... | 289 |
| com.oracle.ouaf.uriSubstitutionVariables.file – URI predefined variables list file | 290 |
| com.oracle.ouaf.uriValidation.enable – Enable Restriction of URLs..... | 290 |
| com.oracle.ouaf.whitelist.file – Oracle URL whitelist filename and location | 291 |
| com.oracle.XPath.LRUSize - XPath Cache..... | 291 |
| com.oracle.XPath.flushTimeout - XPath Flush..... | 291 |
| com.splwg.batch.cluster.jvmName - Online Batch JVM Name..... | 292 |
| com.splwg.batch.scheduler.daemon - Online Submission Daemon..... | 292 |
| com.splwg.grid.executionMode - Online Execution Mode..... | 292 |
| com.splwg.grid.online.enabled - Online Batch Server | 293 |
| com.splwg.grid.distThreadPool.threads.DEFAULT - Online Submission Threadpool | 293 |
| com.splwg.schema.newValidations.F1 - Set behavior of validations..... | 293 |
| jmx.remote.x.access.file - JMX Access Control | 294 |
| jmx.remote.x.password.file - JMX Password Control | 294 |
| oracle.ouaf.jmx.remote.startPort - JMX Override Port | 294 |
| ouaf.accessiblity.features - Enable Accessibility features | 295 |
| ouaf.attachment.disableSpecialCharCheckOnFileName - Disable Special Characters on file name.... | 295 |
| ouaf.batch.onlineLogDir - Online Line Batch Log Directory | 295 |
| ouaf.batch.onlineLogDir.retain – Retain Online Line Batch Log Files | 296 |
| ouaf.database.session.setInstallationTimeZone - Set Time Zone for sessions | 296 |
| ouaf.flush.jms.disabled - Global flush supported | 296 |
| ouaf.flush.jms.connection - Connection Factory for Global Flush | 297 |
| ouaf.flush.jms.requestTopic - Global Flush Request Topic..... | 297 |
| ouaf.flush.jms.responseTopic - Global Flush Response Topic..... | 297 |
| ouaf.runtime.compatibility.enablePrecisionScaleOnMoney - Money Processing | 298 |
| ouaf.jmx.com.oracle.ouaf.ws.mbeans.WSFlushBean - WS Cache Flush..... | 298 |
| ouaf.jmx.com.splwg.base.support.management.mbean.JVMInfo - JVM Information | 298 |
| ouaf.jmx.com.splwg.base.web.mbeans.FlushBean - Manage Cache | 299 |
| ouaf.list.buffer.limit - List Iteration limit..... | 299 |
| ouaf.messaging.transformerfactory - Tranformer | 299 |
| ouaf.shortcut.ignore.altKeys - Disable Hot Keys..... | 300 |
| ouaf.timeout.business_object.default - Set Default Business Object Timeout | 300 |
| ouaf.timeout.business_object.<bocode> - Set Specific Business Object Timeout | 300 |
| ouaf.timeout.business_service.default - Set Default Business Service Timeout | 301 |
| ouaf.timeout.business_service.<bocode> - Set Specific Business Service Timeout..... | 301 |
| ouaf.timeout.query.default - Set Default Query Timeout..... | 302 |
| ouaf.timeout.query.<zonecode> - Set Specific Query Timeout | 302 |
| ouaf.timeout.script.default - Set Default Script Timeout..... | 303 |
| ouaf.timeout.script.<scriptname> - Set Specific Script Timeout | 303 |
| ouaf.timeout.service.default - Set Default Service Timeout | 304 |
| ouaf.timeout.service.<service> - Set Specific Service Timeout..... | 304 |

| | |
|---|-----|
| ouaf.web.appViewer.coreAppURL - AppViewer callback URL | 304 |
| ouaf.web.disablePreload - Disable Screen Preload | 305 |
| ouaf.web.help.coreAppURL - Help callback URL | 305 |
| ouaf.web.helpRootURL - Help URL | 306 |
| ouaf.web.JspHeader.maxAge - JSP Age | 306 |
| ouaf.web.preloadAllPages - Preload Scope | 306 |
| ouaf.web.useOHW - Use Oracle Help for Web engine | 307 |
| ouaf.weblogic.enableDiagnostics - Enable ECID tracking | 307 |
| ouaf.webservices.security.defaultPolicy – IWS customized security policy | 307 |
| ouaf.ws.defaultUser - Default User for Web Services | 308 |
| ouaf.ws.deploy.appxml.file - Application XML file for Inbound Web Services | 308 |
| ouaf.ws.deploy.ear - EAR file name for Inbound Web Services | 308 |
| ouaf.ws.deploy.contextpath - Context Path for Inbound Web Services | 309 |
| ouaf.ws.pageService.annotations - Security Annotation for XAI Legacy services | 309 |
| ouaf.ws.superusers - Effective Web Services Users | 309 |
| ouaf.ws.tools.artifact.generated.buildfile - Build File | 310 |
| ouaf.ws.tools.artifact.generated.sourcedir - Location of generated source | 310 |
| ouaf.ws.tools.artifact.resource.dir - Location of resources | 310 |
| ouaf.ws.tools.artifact.schema.dir - Location of Schemas | 311 |
| ouaf.ws.tools.artifact.xsl.dir - Web Service XSL Location | 311 |
| ouaf.ws.tracing - Inbound Web Services Tracing | 312 |
| ouaf.ws.xslDirOverride - Override XSL Location | 312 |
| ouaf.xai.HTTPBasicAuthPasswordEnc - Default XAI Password | 312 |
| ouaf.xai.HTTPBasicAuthUser - Default XAI User | 313 |
| spl.csv.delimiter.useFromDisplayProfile - List CSV delimiter | 313 |
| spl.ejbContainer.contextFactory - Context Factory for Business Application Server | 313 |
| spl.ejbContainer.password - Password for Business Application Server | 314 |
| spl.ejbContainer.url - Business Application Server URL | 314 |
| spl.ejbContainer.user - Business Application Server user | 314 |
| spl.geocodeDatasource.contextFactory - GIS Context Factory | 315 |
| spl.geocodeDatasource.password - GIS Connection Password | 315 |
| spl.geocodeDatasource.url - GIS Data Source | 315 |
| spl.geocodeDatasource.user - GIS Connection User | 316 |
| spl.mwm.abr.contextFactory - ABR Context Factory | 316 |
| spl.mwm.abr.password - ABR Connection Password | 316 |
| spl.mwm.abr.timeout - Timeout for ABR | 317 |
| spl.mwm.abr.url - ABR URL | 317 |
| spl.mwm.abr.userid - ABR Connection User | 317 |
| spl.mwm.scheduler.abr.maxProcessingTime - Processing Time for ABR | 318 |
| spl.mwm.scheduler.abr.minRequests - Initial Scheduler Pool Size | 318 |
| spl.mwm.scheduler.cleane.interval - Scheduler Cache Clense | 318 |
| spl.mwm.scheduler.connectionTimeout - Scheduler Connection Timeout | 318 |
| spl.mwm.scheduler.maintenanceCycleTime - Scheduler Integrity Check | 319 |
| spl.mwm.scheduler.mapDir - Map Directory | 319 |
| spl.mwm.scheduler.nodeId - Scheduler Node | 319 |
| spl.runtime.mwm.scheduler.ipcStartPort - IPC Start Port | 320 |
| spl.runtime.compatibility.uiMapDisableInputValue - Map Defaults | 320 |
| spl.runtime.compatibility.uiMapDropdownSelectFirstValue - Dropdown defaults | 320 |
| spl.runtime.compatibility.uiMapDisableTitle - Screen Title Defaults | 321 |
| spl.runtime.compatibility.uiMapDisableGenerateUniqueHtmlIDs - Map Ids | 321 |
| spl.runtime.customSQLSecurity – Enable SQL Whitelist | 322 |
| spl.runtime.disableCSRFProtection - Disable Security Tokens | 322 |
| spl.runtime.envIRON.init.dir - Location of Configuration Files | 322 |
| spl.runtime.envIRON.isWebExpanded - Expanded Mode | 323 |
| spl.runtime.envIRON.SPLEBASE - Location Of Software | 323 |
| spl.runtime.initiatize.waittime - Start Time Delay | 323 |
| spl.runtime.envIRON.SPLOUTPUT - Location Of Output | 324 |

| | |
|--|------------|
| spl.runtime.management.batch.cluster.url - Batch Cluster URL..... | 324 |
| spl.runtime.management.connector.url.default - JMX URL..... | 324 |
| spl.runtime.management.rmi.port - JMX Port..... | 325 |
| spl.runtime.options.isFCFEnabled - RAC FCF enabled | 325 |
| spl.runtime.options.onsserver=nodes - ONS Configuration | 326 |
| spl.runtime.options.allowSystemDateOverride - Set Testing Date..... | 326 |
| spl.runtime.options.createSimpleWebAppContextEnable - Web Cache Loading..... | 326 |
| spl.runtime.options.isDevelopmentMode - Development Mode | 327 |
| spl.runtime.oracle.statementCacheSize - Java SQL Cache Size..... | 327 |
| spl.runtime.performSignedNumberValidation.<product> - Number Validation..... | 328 |
| spl.runtime.service.extraInstallationServices - Installation Service | 328 |
| spl.runtime.sql.highValue - SQL High Value..... | 328 |
| spl.runtime.utf8Database - UTF8 Support | 329 |
| spl.serviceBean.jndi.name - JNDI Name for Business Application Server | 329 |
| spl.tools.loaded.applications - Loaded product | 329 |
| hibernate.properties | 330 |
| hibernate.cache.use_second_level_cache - Use Cache | 330 |
| hibernate.connection.datasource - JNDI Data Source..... | 330 |
| hibernate.connection.driver_class - Connection Driver Class..... | 330 |
| hibernate.connection.password - Database User Password | 331 |
| hibernate.connection.provider_class - Connection Provider | 331 |
| hibernate.connection.release_mode - Connection Release..... | 331 |
| hibernate.connection.url - JDBC URL..... | 332 |
| hibernate.connection.username - Database User | 332 |
| hibernate.dialect - Dialect Used..... | 332 |
| hibernate.jdbc.batch_size - JDBC2 Update Support..... | 333 |
| hibernate.jdbc.fetch_size - Solution Set Size | 333 |
| hibernate.max_fetch_depth - Join Fetch Depth | 333 |
| hibernate.query.factory_class - HQL Parser..... | 334 |
| hibernate.query.substitutions - Literal Mappings..... | 334 |
| hibernate.show_sql - Debug Mode | 334 |
| hibernate.transaction.factory_class - Transaction Class | 335 |
| hibernate.ucp.connection_wait_timeout - UCP Connection Timeout..... | 335 |
| hibernate.ucp.inactive_connection_timeout - Connection Inactivity Timeout..... | 335 |
| hibernate.ucp.jmx_enabled - Enable UCP JMX..... | 336 |
| hibernate.ucp.max_size - Maximum Pool Size | 336 |
| hibernate.ucp.max_statements - SQL Buffer size | 336 |
| hibernate.ucp.min_size - Minimum Pool Size..... | 337 |
| submitbatch.properties | 337 |
| com.oracle.ouaf.system.keystore.alias - KeyStore Alias | 337 |
| com.oracle.ouaf.system.keystore.file - KeyStore File | 337 |
| com.oracle.ouaf.system.keystore.hmac_key_alias - KeyStore Mode | 338 |
| com.oracle.ouaf.system.keystore.mode - KeyStore Mode..... | 338 |
| com.oracle.ouaf.system.keystore.padding - KeyStore Padding Algorithm | 338 |
| com.oracle.ouaf.system.keystore.passwordFileName - KeyStore File..... | 339 |
| com.oracle.ouaf.system.keystore.type - KeyStore Type..... | 339 |
| com.ouaf.batch.jvmoptions - JVM Options | 339 |
| com.splwg.grid.executionMode - Mode of Execution | 340 |
| com.splwg.batch.submitter - THIN Submitter | 340 |
| com.splwg.batch.submitter.batchCd - Batch Code..... | 340 |
| com.splwg.batch.submitter.distThreadPool - Threadpool..... | 341 |
| com.splwg.batch.submitter.languageCd - Language Code..... | 341 |
| com.splwg.batch.submitter.maximumCommitRecords - Commit Interval..... | 341 |
| com.splwg.batch.submitter.promptForValues - Prompt for Values | 342 |
| com.splwg.batch.submitter.processDate - Business Date | 342 |
| com.splwg.batch.submitter.rerunNumber - Re Run Number | 342 |
| com.splwg.batch.submitter.softParameter - Job Parameters | 343 |

| | |
|--|------------|
| com.splwg.batch.submitter.threadCount - Thread Limit | 343 |
| com.splwg.batch.submitter.threadNumber - Thread Number | 344 |
| com.splwg.batch.submitter.traceProgramEnd - Trace Program End..... | 344 |
| com.splwg.batch.submitter.traceProgramStart - Trace Program Start..... | 344 |
| com.splwg.batch.submitter.traceSQL - Trace SQL Statements | 345 |
| com.splwg.batch.submitter.traceStandardOut - Trace Output Statements..... | 345 |
| com.splwg.batch.submitter.userId - Security user | 346 |
| spl.runtime.batch.L2CacheMode - L2 Cache control | 346 |
| tangosol.coherence.distributed.localstorage - Cache Mode | 347 |
| tangosol.coherence.role - Submitter Role..... | 347 |
| threadpoolworker.properties | 347 |
| com.oracle.ouaf.batch.scriptDir - base scripts directory for infrastructure job..... | 347 |
| com.oracle.ouaf.system.keystore.alias - Threadpoolworker KeyStore Alias | 348 |
| com.oracle.ouaf.system.keystore.file - Threadpoolworker KeyStore File | 348 |
| com.oracle.ouaf.system.keystore.hmac_key_alias - Threadpoolworker KeyStore Mode | 348 |
| com.oracle.ouaf.system.keystore.mode - Threadpoolworker KeyStore Mode | 349 |
| com.oracle.ouaf.system.keystore.padding - Threadpoolworker KeyStore Padding Algorithm | 349 |
| com.oracle.ouaf.system.keystore.passwordFileName - Threadpoolworker KeyStore File | 349 |
| com.oracle.ouaf.system.keystore.type - Threadpoolworker KeyStore Type | 350 |
| com.oracle.ouaf.system.truststore.file - Location Of Truststore | 350 |
| com.oracle.ouaf.system.truststore.passwordFileName - Location Of Truststore Password file..... | 350 |
| com.oracle.ouaf.truststore.type - Type Of Truststore..... | 351 |
| com.ouaf.batch.disableAutoFlush – Auto flush batch L2 cache | 351 |
| com.ouaf.batch.flushIntervallnSeconds – Batch L2 cache auto flush interval..... | 351 |
| com.ouaf.batch.jvmoptions - Threadpool JVM Options | 351 |
| com.splwg.grid.executionMode - Threadpool Mode of Execution..... | 352 |
| com.splwg.grid.distThreadPool.threads - Threadpool Thread Limit..... | 352 |
| com.splwg.batch.scheduler.daemon - Scheduler Daemon | 353 |
| jmx.remote.x.access.file - JMX Access Control | 353 |
| jmx.remote.x.password.file - JMX Password Control | 353 |
| oracle.ouaf.jmx.remote.maxPortsToTry - Ports to Attempt to connect to..... | 354 |
| oracle.ouaf.jmx.remote.startPort - Override Batch JMX Port..... | 354 |
| ouaf.batch.onlineLogDir - Online Line Batch Log Directory | 355 |
| ouaf.batch.onlineLogDir.retain – Retain Online Line Batch Log Files | 355 |
| spl.runtime.batch.disableDeferredKeyInserts - Deferred Key Inserts..... | 355 |
| spl.runtime.batch.L2CacheMode - Threadpoolworker L2 Cache control | 356 |
| spl.runtime.management.connector.url.default - Batch JMX URL | 356 |
| spl.runtime.management.rmi.port - Batch JMX Port..... | 357 |
| spl.runtime.options.isDevelopmentMode - Development Mode | 357 |
| tangosol.coherence.distributed.localstorage - DistributedCache Storage | 358 |
| tangosol.coherence.distributed.threads - Internal Cache Threads..... | 358 |
| tangosol.coherence.invocation.threads - Internal Invocation Threads | 358 |
| tangosol.coherence.localhost - Unicast Host Address..... | 359 |
| tangosol.coherence.localport - Unicast Port..... | 359 |
| tangosol.coherence.localport.adjust - Unicast Port Adjustment | 359 |
| tangosol.coherence.role - Threadpoolworker Role..... | 360 |
| coherence-cache.config.xml | 360 |
| tangosol-coherence-override.xml | 360 |

Preface

Introduction

Welcome to the Oracle Utilities Market Settlements Management Server Administration Guide for Version 2.3.0.2. 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 V4.4.0.2.0.

Note: For publishing purposes, the Oracle Utilities Market Settlements Management 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 background processing, aka batch, aspects of the Oracle Utilities Application Framework for completeness only. Sections covering the background processing component are marked with a **BATCH** graphic.*

*Note: This document now covers aspects of the mobile framework used for mobile based products. 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](#). 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 Market Settlements Management Installation Guide*
- *Oracle Utilities Market Settlements Management Quick Installation Guide*
- *Oracle Utilities Market Settlements Management DBA Guide*

- *Oracle Utilities Market Settlements Management Security Guide*

These documents are available from [Oracle Software Delivery Cloud](#).

Whitepapers

A number of supplemental whitepaper are available for the Oracle Utilities Application Framework based products from [My Oracle Support](#). At the time of publication¹ the following whitepapers are available on specific topics:

| Whitepaper | Doc Id |
|--|---------------------------|
| Technical Best Practices | 560367.1 |
| Performance Troubleshooting Guideline Series | 560382.1 |
| Software Configuration Management Series | 560401.1 |
| Security Overview | 773473.1 |
| LDAP Integration | 774783.1 |
| Interface Integration Overview | 789060.1 |
| Single Sign On Integration | 799912.1 |
| Architecture Guidelines | 807068.1 |
| Batch Best Practices | 836362.1 |
| What's New In Oracle Utilities Application Framework | 1177265.1 |
| Database Vault Integration | 1290700.1 |
| BI Publisher Integration Guidelines | 1299732.1 |
| Oracle SOA Suite Integration | 1308161.1 |
| Oracle WebLogic JMS Integration | 1308181.1 |
| Oracle Identity Management Suite Integration | 1375600.1 |
| Advanced Security Overview | 1375615.1 |
| Integration Reference Solutions | 1506855.1 |
| Oracle Service Bus Integration | 1558279.1 |
| Using Oracle Text for Fuzzy Searching | 1561930.1 |
| Audit Vault Integration | 1606764.1 |
| Private Cloud Planning Guide | 1643845.1 |
| Migrating from XAI to IWS | 1644914.1 |
| ILM Planning Guide | 1682436.1 |
| ConfigTools Best Practices | 1929040.1 |

¹ Refer to My Oracle Support for any additional whitepapers

| Whitepaper | Doc Id |
|---|---------------------------|
| Oracle Utilities Application Framework - Batch Scheduler Integration | 2138193.1 |
| Implementing Oracle In-Memory Option in Oracle Utilities Application Framework Products | 2404696.1 |
| Using Groovy Script in Oracle Utilities Applications | 2427512.1 |

Architecture

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

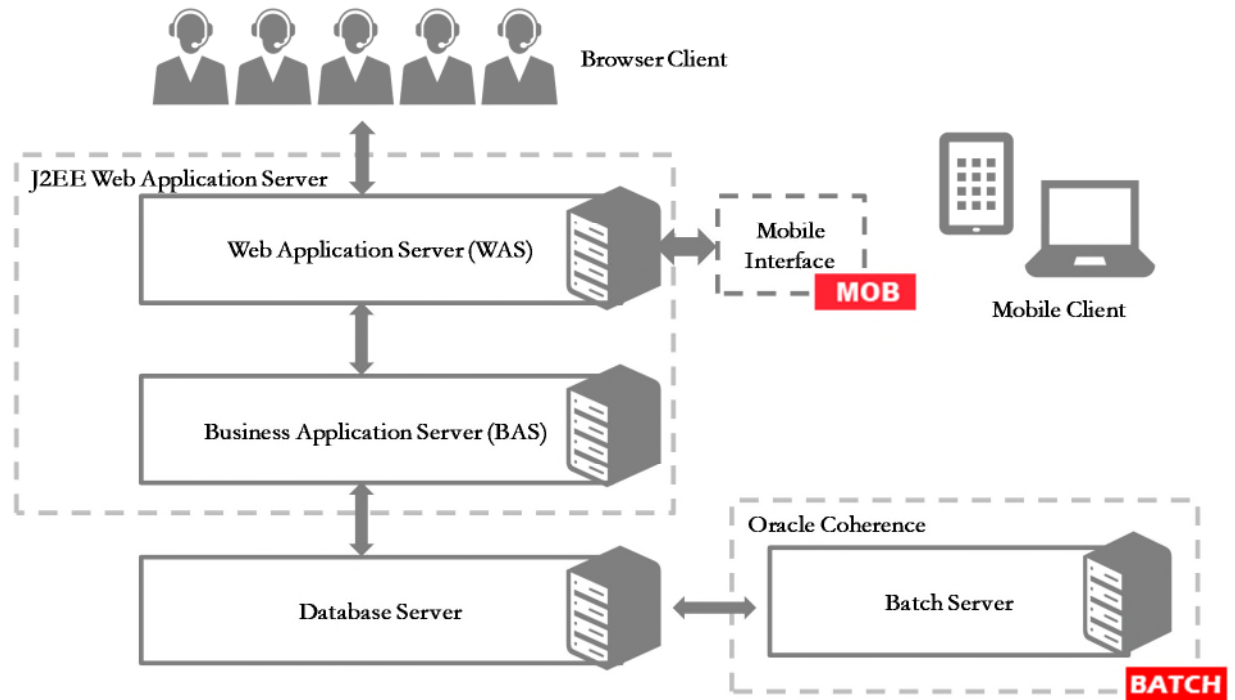


Figure 1 – Product Architecture.

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 HTTPS protocol across a TCP/IP network. Non communications (HTTP) 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 JEE Web application server documentation for proxy support instructions.

Note: By default SSL is enabled with minimal facilities, by default for all new installations. Refer to the SSL documentation provided with your JEE Server to enable SSL to your individual needs.

Note: The default SSL certificate provided by Oracle is recommended for demonstration and development purposes only. It is highly recommended that customers purchase their own certificates for use for SSL.

- **Mobile Device Terminal (MDT)/Mobile Interface** – In some products the Mobile framework is deployed (into Web Application Server ear file) 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 JEE compliant Web application server (Refer to the [Supported Platforms](#) section of the installation guide for JEE 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 JEE 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 JEE compliant Web application server (Refer to the Supported Platforms section of the installation guide for JEE 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. 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.
- **Batch Server** - In some cases, processes need to be processed in bulk and in the background. The Batch Server is a set of JVM's clustered using Oracle Coherence². This feature supports multi-threaded background processes processing records in bulk across a cluster. **BATCH**

² The Oracle Coherence implementation is embedded with the product and is restricted to the background processing component only. Implementation of other Oracle Coherence features beyond that provided with the integration requires the purchase of a full Oracle Coherence license.

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.

Note: In past releases, Adobe/Apache Flash Player was required for displaying and interacting with graphs. This is no longer required.

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.

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 JEE set of Web applications that are housed in a JEE 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 JEE set of business applications that are housed in a JEE 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.

Mobile Application

The product has been enhanced to provide a framework to support integration with mobile devices. The mobile application consists of the Oracle Utilities Mobile library (OUML) of APIs that supports the various application layers responsible for business functionality and communicates with the application server through REST APIs.

The mobile framework supports the following:

- **IOS runtime** – Mobile application/runtime that can run on IOS platform.
- **Android runtime** – Mobile application/runtime that can run on Android platform.

Note: This release of the Mobile Framework is restricted to be used for intranet applications and used for product provided content only.

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.

Note: Refer to the DBA Guide for your product for specific facilities supported by the Database Server.

Batch Server **BATCH**

The product is known for its online (or foreground) processing (a.k.a. *online* processing) but one of the major features of the product is its set of background processes. Background processing is a major part of the product with numerous background processes supplied as *standard*.

The easiest way to understand the concept behind background processing is to think that background processing is like a *super efficient user* that operates on a batch of objects. That is why background processing is commonly called *Batch*.

Note: For publishing purpose the term batch may also be used to denote background processing in this document.

Online typically operates on one object at a time, initiated by an online user or a Web Service call, where batch can operate on one or more objects (also known as a set of objects) at a time, initiated using a number of technologies.

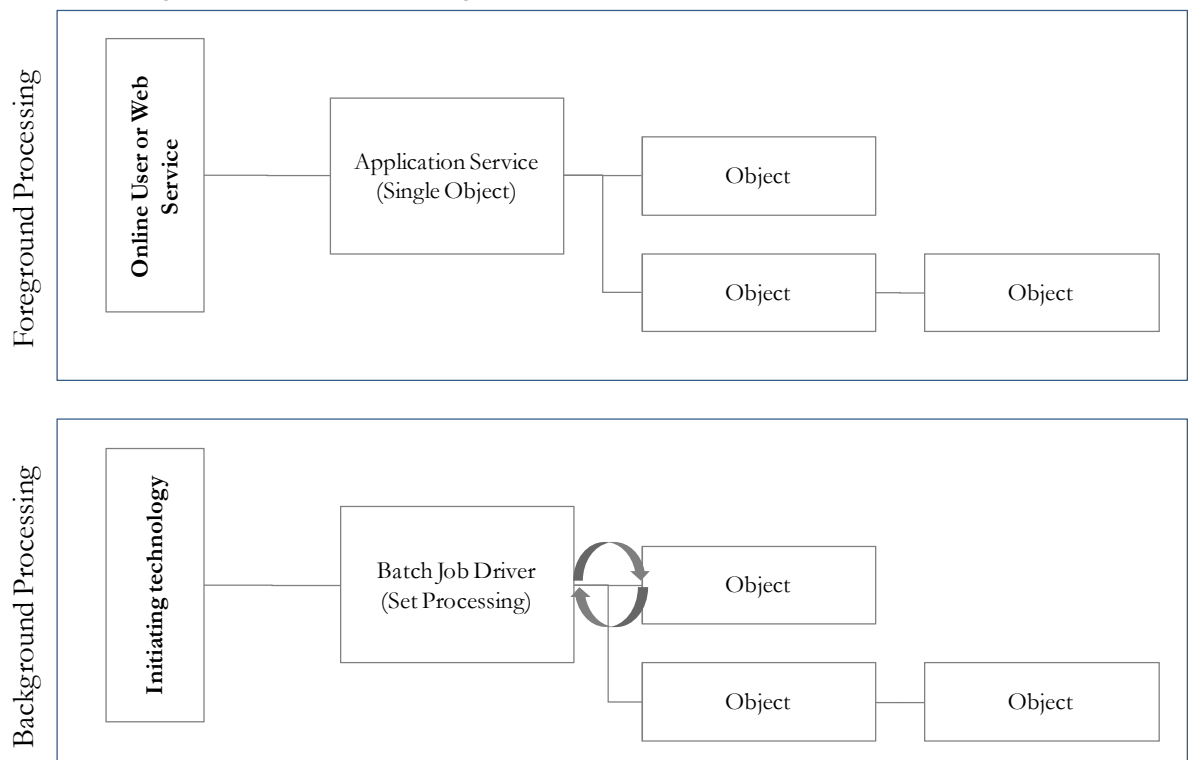


Figure 2 – Batch Processing Architecture

The main reasoning behind the *super efficient user* is that each background process consists of a driver object that identifies the set of valid objects to process and then processes each object through the same business objects that the online uses. For example, the **BILLING** driver determines which accounts are eligible to be billed according to business calendar and then passes each account to the rate object to produce a bill. Contrast this with online bill generation, where the user identifies the account manually, and then that single account is passed to the same rate object to be billed. The background process can call more than one object during the duration of the background process.

For the batch process, all of the database access and object access (including access to business objects, algorithms, user exits (server side only) etc) is done through the Oracle Utilities Application Framework.

Background processing and the Architecture **BATCH**

The Background Processing component is run within the Oracle Utilities Application Framework and is associated typically with the Business Application Server. It is not associated with the Web Application Server and does not require the Web Application Server to be active to operate. The only component other than product that the background processing component requires is the database server (or tier).

Depending on the initiation method employed the background processing component uses a standalone copy of the product to perform access to the database and business objects and its own copy of the same business objects used by the Business Application Server.

Essentially the background processing has its own resources (Java Virtual machines (JVMs), connection pools) independent of the rest of the architecture and can therefore be run on the same hardware as the rest of the architecture or on dedicated hardware.

Batch Architecture Overview **BATCH**

The background processing component of the architecture is a scalable clustered batch approach that allows implementation, using configuration, to spread the batch load across a clustered based architecture.

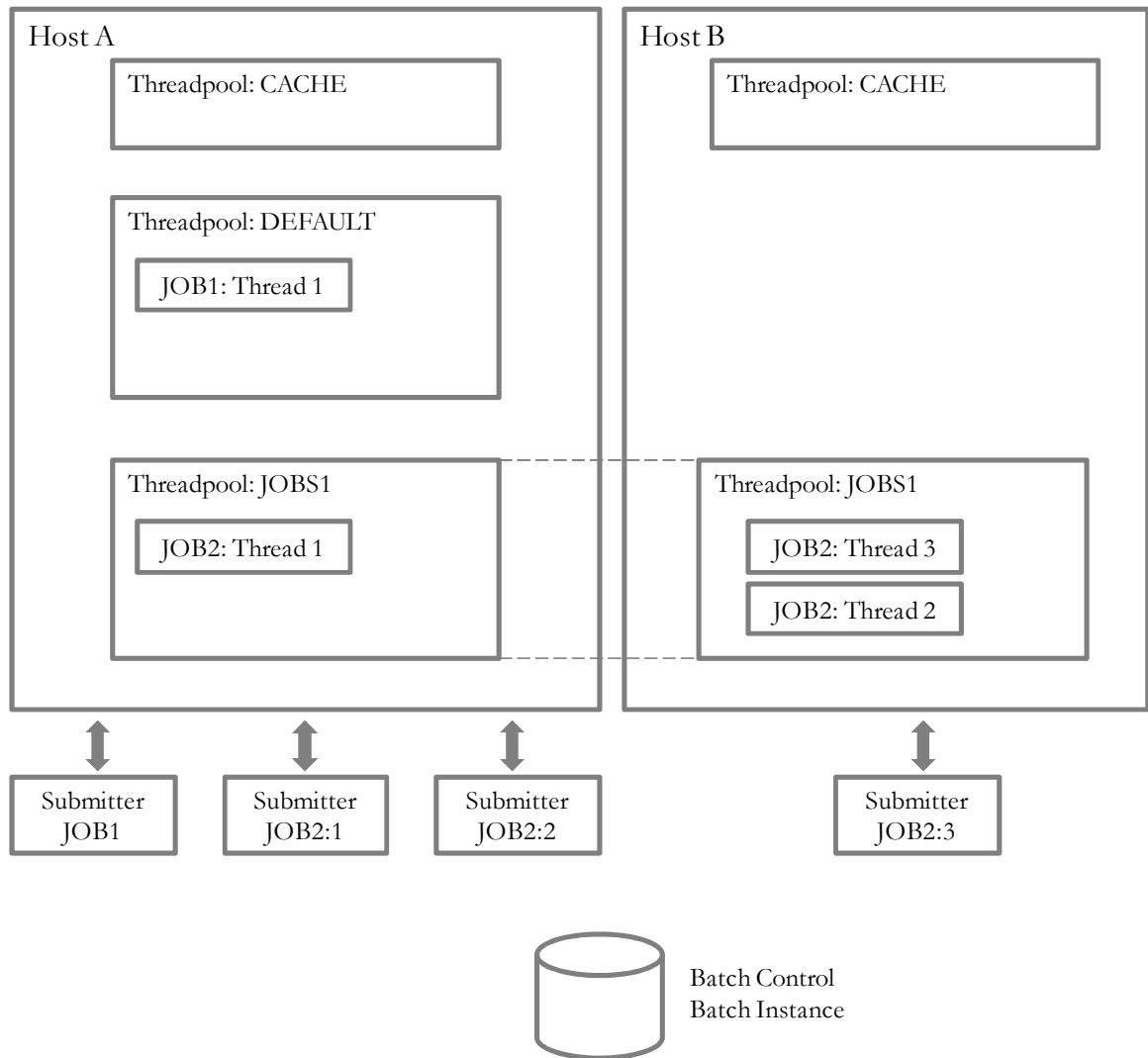


Figure 3 – Batch Architecture Overview

The architecture has the following features:

- A background process cluster is defined using the inbuilt Oracle Coherence³ configuration. This cluster defines the group of hosts, either communicating via unicast or multicast, which will execute background processing threads.
- Within the cluster, the configuration defines threadpoolworkers (also known as *workers*). These are running Java Virtual machines where the batch processes are executed. Threadpoolworkers are named to segregate batch threads and can be clustered across hosts defined in the Coherence cluster. Threadpoolworkers are defined with a maximum limit on the number of threads supported per instance of the threadpool.
- Background processes are submitted (known as *submitters*) to the named threadpools either at a particular threadpool/host instance or automatically distributed across a threadpool across the cluster.
- A background process is defined within the meta data as a [Batch Control](#) which defines the physical characteristics of the job as well as any configurable parameters of the

³ The Oracle Coherence implementation within the product uses a subset of the Oracle Coherence feature set, specifically Named (or Job) Caches.

background process.

- When a background process is executed, a number of threads (*1 to many*) are specified in the execution. Each thread will create *Batch Instance* records to track the progress of the job internally (exposed via the *Batch Run Tree* transaction within the product).
- Oracle Coherence is used, internally, to manage the communication between the cluster and all the members (such as workers and submitters) within that cluster. If a submitter or workers completes the thread or aborts then the cluster is informed and synchronized to take this into account. For instance, in the case of a status change (Completion or abort) the *Batch Instance* record is updated to reflect the state of the job.

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 JEE Web application server. This may be in a single server or across a cluster of servers.
- The business applications deployed in a JEE Web application server. This can be the same physical JEE 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](#) (Doc Id: [560401.1](#)) on [My Oracle Support](#).

Environment Identifier

As part of an installation an environment identifier is generated for the installation. This identifier is used by the technical utilities to manage the components and servers across the architecture. The environment identifier is used in the following ways:

- The Oracle Application Management Pack for Oracle Utilities uses the environment identifier to link child targets to an Utilities Environment target. This is used for monitoring and managing disparate targets (within a single host or across multiple hosts).
- It is used for tracking purposes only outside of the Oracle Application Management Pack for Oracle Utilities.

This feature is used in conjunction with the [Server Roles](#) facility.

Server Roles

One of the features of the installation and configuration is the ability to design architecture roles to a single installation (i.e. a product home). This allows the product to support specific

roles per installation or a combination of roles for a single installation for flexibility. There are four roles available to be used:

| Role | Usage |
|--------------------|--|
| batch | This installation will use the batch server functionality |
| integration | This installation will use the webservice functionality |
| online | This installation will use the online functionality |
| mobile | This installation will use the mobile server functionality |

Selection of each of the roles will enable or disable certain parameters that are available in the [configureEnv\[.sh\]](#) utility.

Note: Even though the role can be set for individual sets of functionality, all the code and facilities are installed and patched to support post installation role change.

Note: If the role is changed post installation additional parameters may need to be provided.

Once the role is set a number of additional configuration parameters are set in the [ENVIRON.INI](#):

- [SERVER_ROLE_BATCH](#)
- [SERVER_ROLE_INTEGRATION](#)
- [SERVER_ROLE_ONLINE](#)

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.

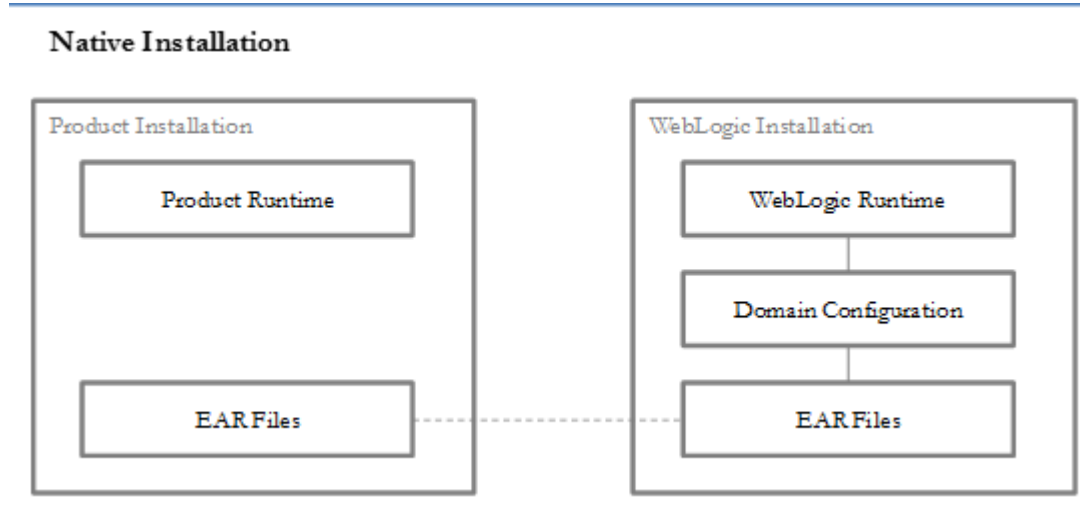
This technical document will refer to the administration userid as **splsys**. If your site uses an alternative userid as the administration userid, substitute that userid value for **splsys**.

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

Starting with OUAF version 4.3.0.5 only native installation support is provided for weblogic. There is no option for embedded mode installation.

In the native mode the product is deployed to the Oracle WebLogic domain location using the Oracle WebLogic deployment tools and managed from the console. In this mode the product runtime for the product is either embedded in the EAR files or referenced indirectly by the EAR file. 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.



Note: For native installation instructions refer to Native Installation Oracle Utilities Application Framework ([Doc Id: 1544969.1](#)) available from My Oracle Support and the installation guides for the product.

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):

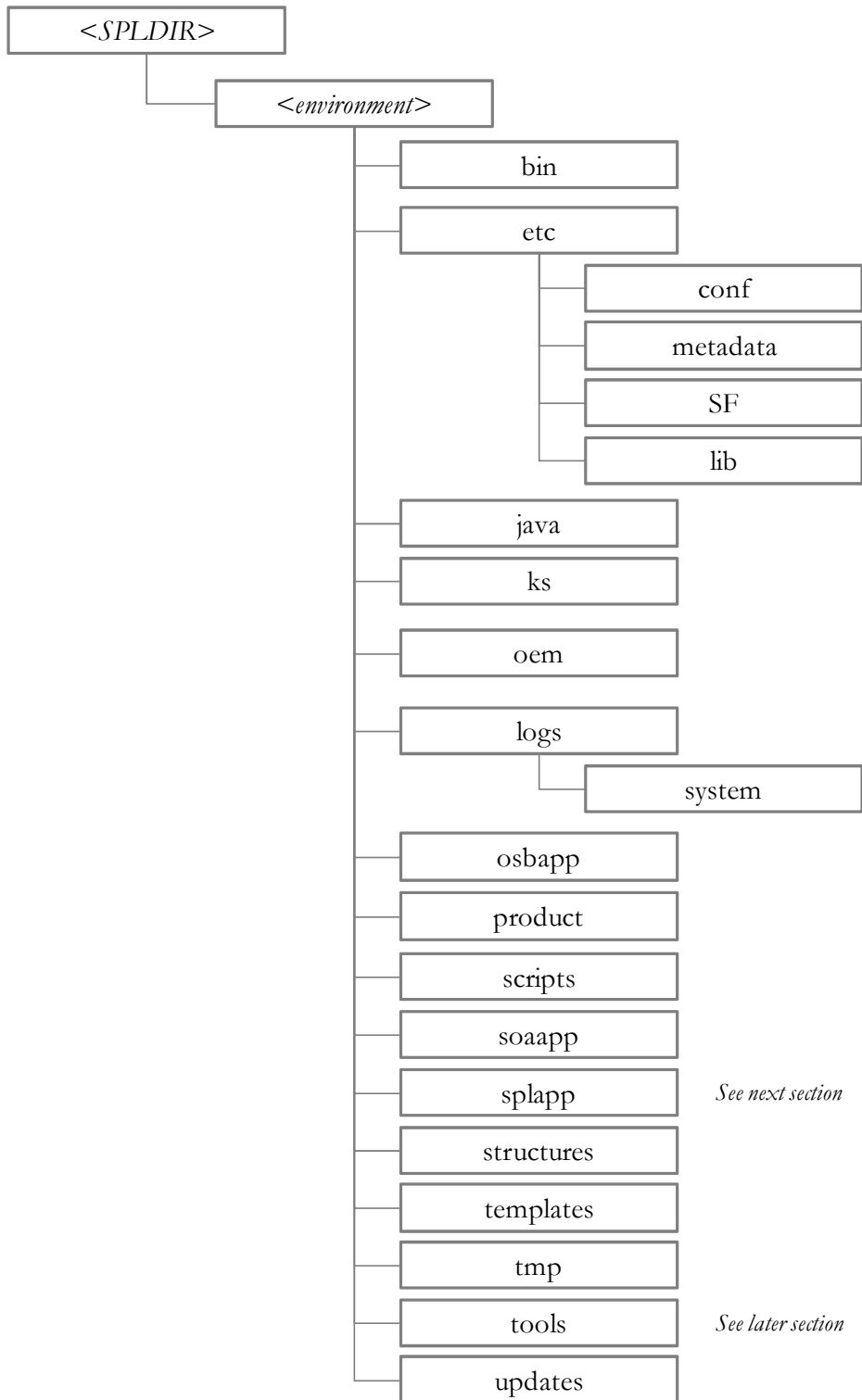


Figure 4 – Software Directory Structure

The following table outlines the typical contents of these directories:

| Directory | Contents |
|------------|---|
| bin | Utilities and commands for operations and configuration. |
| 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. |

| Directory | Contents |
|----------------------------|---|
| java | Location of temporary files for java execution |
| ks | Location of Key Stores |
| oem | Application Management Suite for Oracle Utilities ⁴ generated files |
| logs | Location of product specific log files |
| osbapp | Oracle Service Bus integration (<i>optional</i>) |
| product | Directories containing any bundled software with the product. |
| scripts | Directory containing any implementation specific scripts. |
| soaapp | Oracle SOA integration (<i>optional</i>) |
| splapp | Directories containing the JEE Web Applications (see SPLApp) |
| 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. See Tools . |
| updates⁵ | Location of installed_fixes.txt |

Software (splapp) Directory

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

⁴ Files in this directory are generated by the discovery process used with the Oracle Application Management Pack for Oracle Utilities.

⁵ The **updates** directory only exists after at least one patch has been installed. If the directory does not exist then no patches have been installed on this environment.

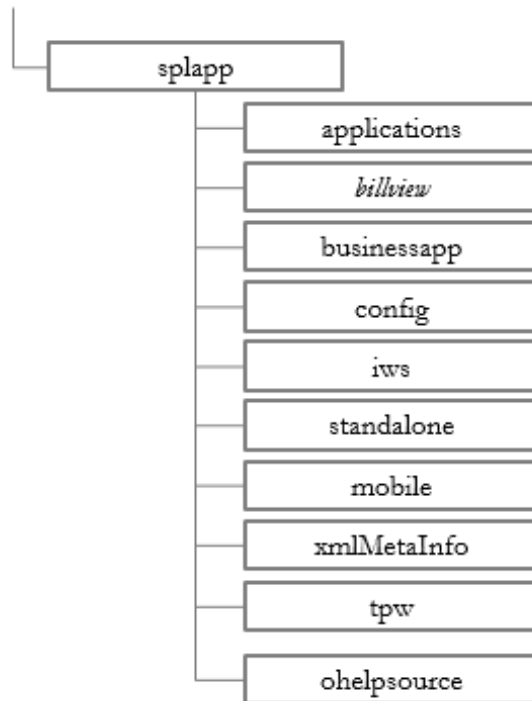


Figure 5 – Software directory structure

| 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 |
| certs | Location of security certificates |
| config | Location of temporary configuration files. |
| iws | Location of Native Web Services |
| mobile | Location of mobile application files |
| ohelpsource | Oracle Help engine source location |
| pending | Internal use only |
| standalone | Location of common Java libraries and the batch component of the product. |
| tpw | Location of threadpool worker specific files. |
| 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.

Tools directory

Note: In past releases the utilities to install patches were included in each patch. This is not the case anymore as they are centralized as part of the installation.

Note: These utilities are not usually executed manually but are part of general installations and automation from Application Management Pack for Oracle Utilities

The tools directory is used for deployment and patching tools used by Application Management Pack for Oracle Utilities and the internal patching utilities. The directories are in the following structure:

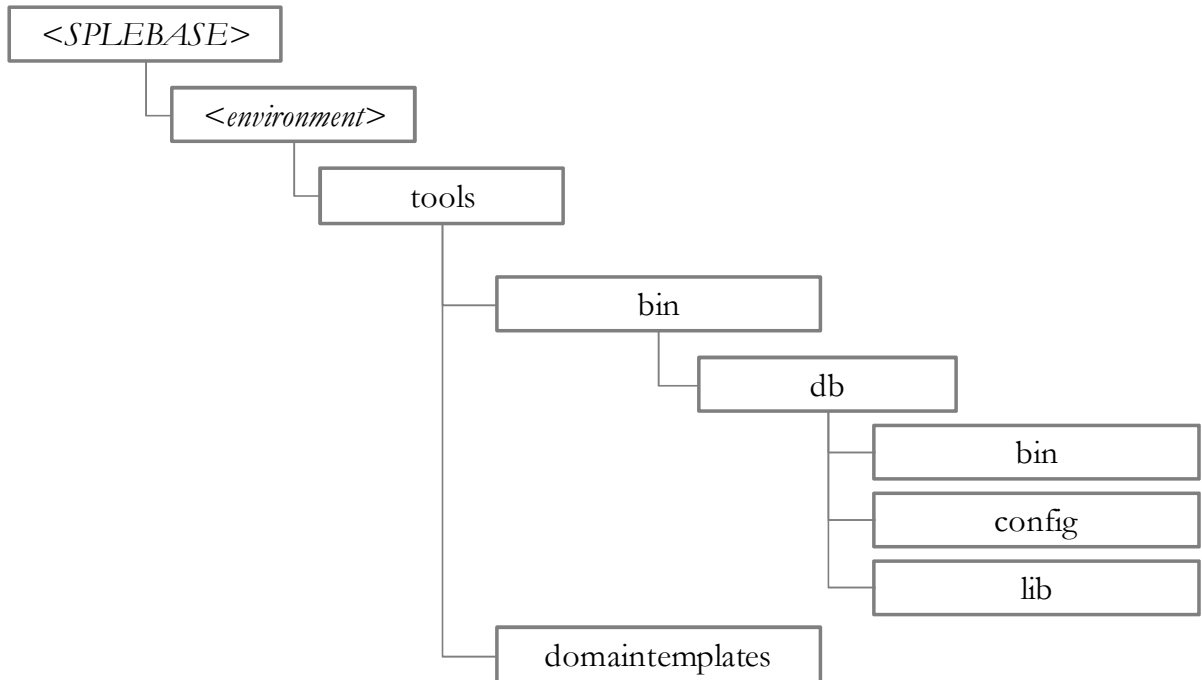


Figure 6 – Tools directory structure

| Directory | Contents |
|------------------------|--|
| bin | General Software patching utilities |
| bin/db | Database Patching utilities |
| bin/db/bin | Binaries for patching |
| bin/db/config | Logging configuration for patching utilities |
| bin/db/lib | Java Libraries for patching utilities |
| domaintemplates | Oracle WebLogic Domain Templates for product |

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 **\$SPLBASE** location:

| Directory/files | Owner | Group | Other |
|-------------------------------------|-------|-------|-------|
| Etc | rw | r | |
| Individual files | rw | r | |
| Logs | rw | r | |
| Tlogs/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):

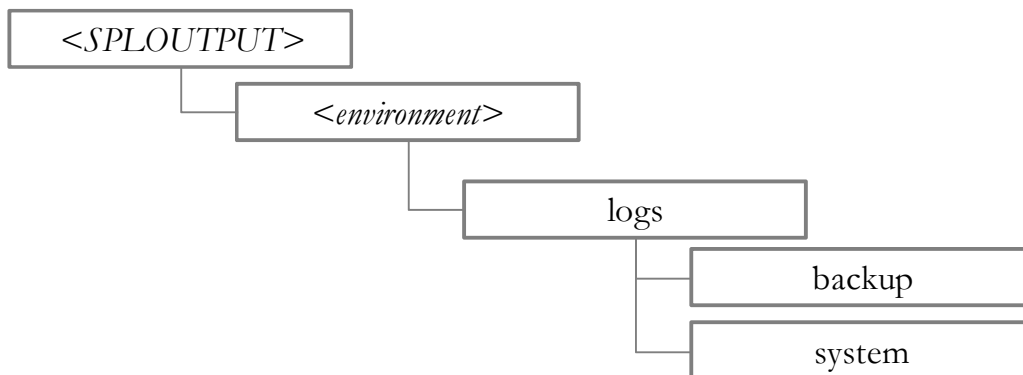


Figure 7 – Output directory structure

| Directory | Contents |
|-----------------------------|--|
| Togs/system | Directory containing application logs files. This is independent of Web application server, Business Application Server and Database Server log files. |
| Togs/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 |
|------------------------------------|---|
| ANT_ADDITIONAL_OPT | Options for <i>ant</i> as per the configuration settings at installation time |
| ANT_HOME | Location of <i>ant</i> build utilities |
| ANT_OPTS | Options for <i>ant</i> for Oracle SDK |
| ANT_OPT_MAX | Maximum memory settings for <i>ant</i> as per the configuration settings at installation time |
| ANT_OPT_MIN | Minimum memory settings for <i>ant</i> as per the configuration settings at installation time |
| CMPDB | Database Type (ORACLE) |
| DBNAME | Database Name |
| ENVFILE | Location and name of environment configuration file |
| HIBERNATE_JAR_DIR | Location of Hibernate Java library |
| HOSTNAME | Name of host (fully qualified) |
| JAVA_HOME | Location of JDK |
| LD_LIBRARY_PATH | Library Path |
| 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 |

| Variable | Usage |
|--------------------------------|--|
| | 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 this value will match ORACLE_HOME . |
| ORACLE_HOME | Location of the DBMS software |
| OSTYPE | Operating System Type |
| PERL5LIB/PERLIB | Location of Perl Libraries |
| PERL_HOME | Location of Perl |
| SERVER_ROLES | Roles enable for this server |
| SERVER_ROLE_BATCH | Whether the batch server is included |
| SERVER_ROLE_INTEGRATION | Whether the integration server is included |
| SERVER_ROLE_ONLINE | Whether the online server is included |
| 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). |
| SPLBASE | 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) |
| SPLSYSTEMLOGS | Location of product specific logs |
| SPLVERSION | Version identifier of product (prefixed with V) |
| SPLVERSION_NUM | Version number of product |
| WEB_IEXPANDED | Whether Web application is expanded or not (not = WAR/EAR files) |
| WEB_SERVER_HOME | Location of Web Application Server software |

| Variable | Usage |
|----------------|--|
| WL_HOME | Location of Oracle WebLogic installation (WebLogic supported platforms only) |

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.*

Note: Additional variable may be created by various operating system and product utilities. The list above consists of the non-temporary variables.

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 JEE Web application server will log Web Application errors or messages to the JEE 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). Some of these logs are further classified into technical and user logs. As the name suggests, the technical logs contain tracing information for SQLs, stack traces of java code and other similar technical traces. Essentially, these are errors that suggest the system is misconfigured or is being run in a way that is not expected and also contain Application Trace Data which is required for diagnostic purposes. Whereas the user log consists of a Business logic messages, Business trace data.

The following are the logs:

- **spl_web-tech.log** - Web application server application messages that are technical in nature.
- **spl_web-user.log** - Web application server application messages that contain user/business data or traces.
- **spl_service-tech.log** – Business Application Server technical messages. If the Business Application Server exists on the same JEE 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-tech.log**.
- **spl_service-user.log** – Business Application Server business logic/trace messages. If the Business Application Server exists on the same JEE 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-user.log**.
- **spl_iws.log**– Web Services Adapter messages.
- **spl_mobile-tech.log** – Technical messages for server side component of mobile application.

- `spl_mobile-user.log` – Business messages for server side component of mobile application.
- Background processing logs are also categorized into technical and business logs and are located in `$SPLOUTPUT` (or `%SPLOUTPUT%` on Windows) for the following:
 - `threadpoolworker.<threadpool>.<datetime>-tech.log` - The threadpoolworker technical log for threadpool `<threadpool>` started at `<datetime>`.
 - `threadpoolworker.<threadpool>.<datetime>-user.log` - The threadpoolworker business log for threadpool `<threadpool>` started at `<datetime>`.
 - `<batch_cd>.<datetime>.THD<thread>.stderr/<batch_cd>.<datetime>.THD<thread>.stdout` - The submitter log for thread `<thread>` of background process `<batch_cd>` started at `<datetime>`.

The format of all logs is as follows:

| Field | Comments |
|------------------------------------|--|
| <code><userid></code> | User ID of transaction (blank or "-" for system generated messages) |
| <code><transactionId></code> | Transaction identifier |
| <code><ECID></code> | Execution Context Identifier (weblogic) |
| <code><date/time></code> | Time of transaction in format YYYY-MM-DD HH24:MM:SS,SSS |
| <code><threadinfo></code> | Information of the thread |
| <code><log level></code> | Logging level enabled |
| <code>(<class>)</code> | Java class generating message (see Javadocs in appViewer) (applicable for tech logs) |
| <code><message></code> | <message contents> |

Sample log entries:

User Application Error logs:

```
SPLRXO - 364812-788-1 2016-11-16 13:08:44,371 [[ACTIVE] ExecuteThread:
'13' for queue: 'weblogic.kernel.Default (self-tuning)'] INFO
(support.changehandlers.ChangeEventHandler) PK1 PK2 combination not found.
Check if the fields are populated. Message Number 11001, 402.
```

Tech Application Error logs:

```
SPLRXO - 364812-788-1 2016-11-16 13:08:44,371 [[ACTIVE] ExecuteThread:
'13' for queue: 'weblogic.kernel.Default (self-tuning)'] INFO
(support.changehandlers.ChangeEventHandler) The following rule was
violated: Characteristic value should be valid :
userCharacteristic:Characteristic value should be valid for entity 'FLUS'
```

Batch Concepts **BATCH**

*Note: The **DISTRIBUTED** mode supported in previous versions of the product had been removed earlier. Customers using this mode should migrate to **CLUSTERED** mode using either a single server configuration (**ss**) or unicast configuration (**wka**) using **bedit**.*

The batch component of the architecture has a number of key concepts to understand.

Batch Clustering Concepts

As mentioned in the previous section, Oracle Coherence is used internally to cluster the resources needed for the batch architecture. This is similar in concept to a cluster used by the JEE Web Application Server for online and web services.

- A cluster is one or more hosts that can execute batch processes. A cluster is named to delineate it from other clusters in your network. Oracle Coherence Named Caches are used to implement the cluster.
- Members of a cluster are hosts, threadpools and submitters. Each member is tracked by the cluster for high availability. At least one member must be active for a cluster to operate.
- Intracluster communication between members of the cluster can be achieved using uni-cast (also known as Well Known Addresses) or multi-cast.
 - In uni-cast implementations, each host and associated port is defined in the cluster. Each cluster member has definitions of other members. This mode is a less dynamic configuration⁶ but optimizes host to host communications for large clusters⁷.
 - In multi-cast (*default*) implementations, a multi-cast address and port are defined across the cluster which members attach to when executing. This mode is dynamic but traffic across nodes can be excessive for large clusters.
- A threadpool is a Java Virtual Machine (JVM) that executes batch threads. It is a long running JVM that batch threads are submitted to. Threadpools are named to group resources across a cluster when using multiple instances of a threadpool.
- A threadpool has a configurable maximum capacity for running batch threads to prevent overloading of JVM's⁸. Running multiple instances of a threadpool (within or across hosts) can increase the limit and implement high availability.
- Clusters can support dedicated cache threadpools. In a cluster, each member can potentially communicate to each other member. In a large network this can cause network overheads that can adversely affect performance. To minimize the impact of network traffic, it is possible to configure a cache threadpool that will act as a central co-ordinator in the cluster to optimize and channel intramember communication.
- Clusters can support administration threadpools. In a cluster, it is possible to check on the status, in real time using JMX, across the cluster by connecting to any node in the cluster. It is possible to setup an administration node that can act as a single point of contact across the cluster to monitor performance, in a similar vein as the administration server in the JEE architecture. These administration threadpools can be used by command line utilities, jconsole, Oracle Enterprise Manager via Application

⁶ As each member must have its own configuration. To add or remove members requires manual synchronization of configuration files.

⁷ Large clusters are a large number of hosts, large number of threadpools and/or large number of threads executing at the same time.

⁸ Threads in background processing are typically heavier in terms of resource usage than typical java threads hence lower thread limits.

Management Pack for Oracle Utilities and/or Coherence Management Pack.

- Submitters are instances of individual background processes or individual threads of background processes that are submitted to a named threadpool. The submitter waits for the thread or background process to complete or abort and reports the status of the thread or background process to the operating system. This latter facility is used by Batch Scheduling tools to determine the status of the background process.
- Submitter can be targeted to specific instances of a threadpool or submitted to the cluster of threadpools. Submitters can execute specific threads or all threads in a single execution.

Batch Controls

Note: The system is delivered with all necessary batch controls for the supplied base background processes.

Background processes within the product must be defined using meta data (as well as physical code) to execute the process in the architecture. Refer to the online help and user documentation for details of the attributes associated with a batch control.

Standard parameters

To standardize all the background processes, the product uses a number of common standard parameters to uniformly provide functionality across all processes. The table below lists all the standard parameters:

| Parameter | Usage |
|---------------------|--|
| Batch code | Code is the unique identifier of the background process |
| Batch thread number | Thread number is only used for background processes that can be run in multiple parallel threads. It contains the relative thread number of the process. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). |
| Batch thread count | Thread count is only used for background processes that can be run in multiple parallel threads. It contains the total number of parallel threads that have been scheduled. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. |
| Batch rerun number | Rerun number is only used for background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run). |
| Batch business date | Business date is only used for background processes that use the current date in their processing. For example, billing using the business date to determine which bill cycles should be downloaded. If this parameter is left blank, the system date is used. |

| Parameter | Usage |
|---------------------|--|
| Commit Interval | Override maximum records between commits. This parameter represents the number of transactions that are committed in each unit of work. This parameter is optional and overrides the background process's Standard Commit between records (each background process's Standard Commit between records is documented in the product documentation). You would reduce these values, for example, if you were submitting a batch process during the day and you wanted more frequent commits to release held resources. You might want to increase these values when a background process is executed at night (or weekends) and you have a lot of memory on your servers. |
| Application Service | This is the application service that is associated to the batch. All the users who are allowed in the application service which is associated with particular batch control will be allowed to submit the batch job. |
| Language | This is the language code used to retrieve messages and format output from background processes. |
| Traces | Trace program at start (Y/N), trace program exit (Y/N), trace SQL (Y/N) and trace output. <ul style="list-style-type: none"> • If trace program start is set to Y, a message is displayed whenever a program is started. • If trace program at exit is set to Y, a message is displayed whenever a program is exited. • If trace SQL is set to Y, a message is displayed whenever an SQL statement is executed. • If trace out is set to Y, message are output from the program at execution points. |

Note: This facility should only be used in development.

Additional Key Parameters

In all batch controls there are two additional parameters that may not be specified in the batch control definition but are available for use within each job:

- **MAX-ERRORS (maxErrors)** - This is the error tolerance for the job. When the number of errored records in a thread for a job has been exceeded that thread is stopped with an appropriate error code. This setting is to prevent large amounts of errors to influence the state of the job.
- **DIST-THD-POOL (threadPool)** - This allows a particular pool to be specified for the execution of the job. This is primarily used with online submission.

- **emailMode** - Defines the status for which an email notification⁹ is sent when a job ends. This does not apply to individual threads but when all threads are complete. Valid Values are:
 - **ERROR** - Send an email when at least one thread ends in error status.
 - **SUCCESS** - Send an email when all threads end successfully.
 - **ALL** (Default) - Always sends an email when all threads end regardless of status.
- **overrideLowIdValue** - Specifies a new low id to use in calculating the range for a thread. The product by default assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 999999999), but this parameter will override the low value. The product calculates each thread's low and high id by slicing the job's range into chunks using the thread number and count. It is typically more appropriate for sequentially allocated Ids to evenly distribute work among threads. The parameter value can be an actual number or it can be set to *auto*. If the parameter is set to *auto*, then the value is set to the lowest current value on the associated database table. The table name is determined from the batch program's query iterator
- **overrideHighIdValue** - Specifies a new high id to use in calculating the range for a thread. The product, by default, assumes that the Id is between 0's (e.g. 000000000) and 9's (e.g. 999999999), but this parameter will override the high id. The product calculates each thread's low and high id by slicing the job's range into chunks using the thread number and count. It is typically more appropriate for sequentially allocated Ids to evenly distribute work among threads. The parameter value can be an actual number or it can be set to *auto*. If the parameter is set to *auto*, then the value is set to the highest current value on the associated database table. The table name is determined from the batch program's query iterator.
- **idRangeOverrideClass** - Provides for a custom class to do thread range calculation. The class name specified must implement interface **com.splwg.base.api.batch.BatchIdRangeOverride**. During batch execution, this override class will be instantiated and the setter methods called to initialize the Ids as required. The low and high getter methods will be called to retrieve the high and low ids to be used for the run
- **numRecordsToFlush** - Defines how frequently to flush the Hibernate cache to prevent high heap consumption and **OutOfMemoryErrors**. In **SingleTransactionStrategy** there are no commits throughout the run, so this parameter can be used to flush the Hibernate cache periodically. This parameter is only applicable to jobs that use **SingleTransactionStrategy**.

*Note: The parameters **overrideLowIdValue** and **overrideHighIdValue** are only applicable to jobs that use **ThreadIterationStrategy**.*

These parameters may be added to the Batch Control or properties files used for running the job.

⁹ This is used for online submission when an email address is provided on the job submission and the email server is configured.

Batch Run Tree

Within the product browser interface there is an ability to monitor the status and outcomes of individual processes. This can be useful for finding out what actually occurred if an error condition occurred. Refer to the online help for more information about this facility.

The Batch Run tree is available to any valid user and is a method to communicate the execution information to the relevant business representatives.

Timed Jobs

Note: Monitor processes are ideal for Timed Jobs.

By default, background processes start and finish when they have processed the records it has selected at the start of execution. In this case, to process more records it is necessary to run the batch process again.

In some business processes, it may be necessary to continually execute a job to process work that is ready at any time. For example, processing payments, meter reads, asset changes or tasks as they appear. This situation is handled as a *Timed Job*. This Batch Control type is configured with a *Timer Interval* (the desired runtime between executions in seconds) and a *Timer Active* flag (to enable and disable the job activity).

When the *Timer Active* flag is set to **true**, the product executes the job for the duration of the *Timer Interval* before executing another instance of the job. This allows the job to run continuously over time, whilst the *Timer Active* flag is set to **true**.

Each instance of the job will start and stop using the following rules:

- The job will only execute when the Timer Active Flag is set to **true**. To the next execution from starting this value should set to **false**.
- The job will execute using the parameters on the Batch Control including the dedicated parameters for Timed Jobs such as *UserID*, *Batch Language* and *Email Address*.
- Once the Timer Interval has been reached, the job will commit after the latest transaction is complete and update the Batch Control is complete or errored (as appropriate).
- The next instance of the process is repeated as above.

Note: Due to varying transaction workloads the time taken may exceed the Timer Interval. This situation can be tracked using Level of Service if desired.

Level Of Service

Note: For backward compatibility purposes, Level of Service is in the disabled state and requires configuration of targets in the Level Of Service algorithm.

One of the features of the background processing architecture is that all executions of a Batch Control are tracked internally. This is typically expressed in the *Batch Run Tree* for information purposes but can also be used to calculate an indicator of the level of service if desired using the *Batch Control - Level Of Service* algorithm.

Algorithms of this type return one of three levels of service:

- **Disabled** - Batch level of service algorithm is not enabled for the batch control or no

data is available for comparison¹⁰.

- **Error** - Indicates that errors were detected. In this case a message is also returned to indicate the issue.
- **Normal** - Indicated when there were no errors or issues detected by the algorithm.

Note: It is possible to write custom Level Of Service algorithms to compare any appropriate business metric.

Explanation of Commit Interval

The Commit interval parameter is a tuneable parameter to affect the impact of the background processes on the other processes running and prevent internal database errors. In most cases using the defaults will satisfy your site requirements. It is also important to understand their impact to ascertain whether any change is required.

During processing of any background process a main *object* is used to drive the process. Each process tables one or more objects to process. For example, billing processes accounts, meter reading processes meter reads or meter events, asset Processing uses assets, workforce processing uses tasks etc. For other processes it is other *objects* that are considered the main object. This status of the object or status of related transaction data is used to determine when a transaction is complete.

For the Commit interval this concept is important as:

- When a certain number of main objects have been processed then a database commit is issued to the database. This number is expressed in the *Commit Interval*. The larger the commit interval the larger the amount of work that the database has to keep track of between commit points.

Note: The number of records in progress utilizes redo log space so the size of the redo logs in the database and commit interval must be compatible. The higher the commit interval the more redo log space is required.

- At any time in a process a commit for objects processed may be caused by the reaching the Commit Interval or the end of the job, whichever comes first.

Explanation of Thread Limit and Thread Number

Note: Optimization discussions for these parameters are covered in the [Batch Best Practices](#) (Doc Id: [836362.1](#)) available from [My Oracle Support](#).

One of the features of the Oracle Utilities Application Framework is the ability to run background processes using multiple threads.

The threading concept in the product is simple. Each thread takes a predetermined slice of the data to work on. The last thread checks if all other threads are finished and updates the status of the batch control records. For example, if you have 10 threads, then each thread takes 1/10th of the work. As each thread is executing it processes its workload and then completes, the last thread executing is responsible for updating the overall process status to indicate completion. Implementing threading means you have to execute a number of background processes with

¹⁰ This can occur for the initial execution only.

an ascending thread number up to the thread limit. For example, if you have a background process with 10 threads, you must run 10 background processes each with a unique thread number between 1 and 10 to complete the background process. Threads can be located on the same machine or different machines. For example, you can run threads 1 to 5 on one machine and threads 6 – 10 on another.

Note: If there is limited data skew in the data then the threads should finish around the same time. If there is some data skew then some threads may finish later than others.

To implement multi-threading when you submit a process:

- Specify a thread limit greater than 1 as a parameter.
- Execute a background process for every thread with a sequential thread number up to an including the thread limit. There are a couple of implementation guidelines with threading:
 - Make sure the number of threads is not excessive. You do not want to flood the CPU's.
 - You must submit a process per thread. In some submission methods this is done automatically and in some it is done manually.
 - Threading will increase throughout BUT it will cause higher than usual resource usage (CPU, Disk etc) as well as higher contention. Excessive threading can in fact cause performance degradation in online as well as background processing. Therefore the number of threads should not be excessive.

Almost all background processes within the product support multiple threads (typically the only background processes that are single streamed are extracts and data loads as they involve reading and writing files).

Explanation of Restart and Rerun

The product allows all background processes to be restarted or rerun as required. During the execution of the background process, restart information per thread is stored within framework, like a *checkpoint*. This checkpoint is performed at the last commit point as dictated by the Commit Interval and/or Timeout value (Time out only applied to Oracle implementations only). When a *commit* is performed, the last commit point is recorded for the execution. If a thread of a background process fails, the database automatically rolls back to the last commit point. The thread can then be restarted from that point automatically or from the start of the data. To indicate the restart, the thread is executed with the same parameters as the original.

Additionally, processes are *re-runnable*. A background process is *re-run able* when you can specify a specific run number to be re-run as required or specify a specific business date to reprocess. Using a rerun number or a previously used business date are all that is required to rerun a process.

Note: Not all background processes use Run number as a run indicator. Refer to the online documentation for which batch processes are re-runnable.

Submission Methods

There are a number of methods for submission supported by the product.

- [Online Submission](#) - This allows testing users to submit batch jobs from the online component. This facility is typically used for non-production environments.

Note: Oracle strongly recommends that online submission is not used for Production environments.

- [Command Line Submission](#) - A number of commands are available to start/stop threadpools and jobs from the command line or from a third party scheduler.

Online Submission

One of the most important useful testing/demonstration facilities of the product is the ability to submit batch processes from the online component of the product. An authorized user can submit any batch process using an online batch submission page.

The on-line batch submission page enables you to request a specific background process to be run. Refer to the online help for a discussion on how to use the online submission function. When submitting a background process on-line, you may override standard system parameters and you may be required to supply additional parameters for your specific background process. After submitting your background process, you may use this page to review the status of the submission.

Basically the following process is used to submit background processes using the online submission method:

- The process to be executed is registered online as to be submitted (or queued). This marks the process execution as *Pending*. When you request a batch process to be submitted from on-line, the execution of the desired background process will result in the creation of a batch run. Just as with background processes executed through your scheduler, you may use the Batch Run Tree page to view the status of the run, the status of each thread, the run-instances of each thread, and any messages that might have occurred during the run.

Note: Your online submission record is assigned a status value so that you may know whether your batch process has been submitted and whether or not it has ended; however, it will not contain any information about the results of the background process itself. You must navigate to the Batch Run Tree page to view this detail.

- A background process is scheduled using the submission daemon that will pickup any *Pending* background process executions and execute them. When you save a record on the batch process submission page, the batch process does not get submitted automatically. Rather, it saves a record in the batch process table. A special background process will periodically check this table for pending records and will execute the batch process. This background process will update the status of the batch process submission record so that a user can determine when their batch process is complete.

Note: At installation time, your system administrator will set up this special background process or configure the scheduler daemon to periodically check for pending records in the batch process submission table. Your administrator will define how often the system will look for pending records in this table.

It should be noted that this special background process only submits one pending batch process submission record at a time. It submits a batch process and waits for it to end before submitting the next pending batch process.

Note: If you request a batch process to be run multi-threaded, the special background process will submit

the batch process as requested. It will wait for all threads to complete before marking the batch process submission record as ended.

During execution the status of the execution in the batch run tree is updated as well as the original submission screen. If you wish the system to inform you when the background process completes, you may supply your email address. The email you receive will contain details related to the batch process's output; similar to the batch process results you would see from the batch run tree.

Note: This assumes that during the installation process, your system administrator configured the system to enable email notification. Your administrator may also override the amount of detail included in the email notification.

To use this facility the following must be enabled:

- The online submission daemon and DEFAULT threadpoolworker is configured using the following parameters:
 - The [BATCHENABLED](#) setting must be set to true.
 - The [BATCHDAEMON](#) setting must be set to true.
 - The [BATCHTHREADS](#) setting must be greater than zero.

Note: The default setting for [BATCHTHREADS](#) is 5 threads. This setting should only be altered if large numbers of testers need to use the DEFAULT threadpoolworker at the same time.

- An external DEFAULT threadpool and daemon is started using the **-d Y** option on the [threadpoolworker](#) or [starttpw.sh](#) command. For example:
`threadpoolworker.sh -p DEFAULT=5 -d Y`

Command Line Submission

At a site implementing the product, the batch processes to be executed to support the business as well as perform expected maintenance on the system needs to be scheduled, managed and executed from a central point. In most sites, this is done by using a third party batch process scheduler that controls the scheduling and execution of any batch processes across a site.

To support the use of such a scheduler with any Oracle Utilities Application Framework based product(s) a number of scripts and related configuration files have been provided to allow the scheduler to execute the process batch processes.

The scripts and configuration files allows for three fundamental facilities that can be used by external scheduling tools:

- The interface is command line based (*it can also be invoked using a java based API see the product javadocs within AppViewer for a description of the interface*) which most external scheduling tools support.
- The command based utilities return a standard return code to indicate the batch process has been successful or has been unsuccessful. Actions dependent on return code within the scheduler can then be configured.
- The logs within the utilities provided are in a common format that can be interrogated by the external scheduler to provide finer grained actions (*especially for unsuccessful executions*).

For additional advice about interfacing external schedulers with the product refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) whitepaper on [My Oracle Support](#).

The following guidelines apply:

- Any command should execute [splenvi ron](#) prior to execution.
- Threadpoolworkers should be started prior to running any batch job threads using the [threadpoolworker](#) or [starttpw.sh](#) command in the scheduler.
- Batch Job threads should be submitted using the [submitjob](#) command.
- If threadpoolworkers need to be shutdown the [jmxbatchclient -k](#) command should be used.

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: This section only applies to the embedded mode installation of the product. Customers using native mode installations should start and stop servers via the Oracle WebLogic console, WLST or Oracle Enterprise Manager.

Command Line Utilities

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

bedit - Batch Edit **BATCH**

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: This facility is only available if "Enable Batch Edit" is set to true in the installation.

To simplify the configuration of batch a new administration facility has been introduced. The **bedit[.sh]** is provided to guide the administrator through the cluster, threadpool and submitter configuration process using templates.

Command Usage:

Linux/Unix:

```
bedit.sh [-h] [[-g] [-t arg] [-b arg|-e arg|-l arg|-c|-s|-w] [file]] | [-A]
```

Windows:

```
bedit [-h] [[-g] [-t arg] [-b arg|-e arg|-l arg|-c|-s|-w] [file]] | [-A]
```

Where:

- h** Show command usage
- h** Show extended help
- g** Generate configuration from the template and exit immediately. The configuration file name may be specified explicitly or implicitly as described under Mutually Exclusive Arguments below.
- t arg** Template to use. Examples: **wka** for Well-Known-Addresses; **mc** for Multicast; **ss** for Single Server. This is optional. The default is determined from **be.prefs**, **be.properties** or **be.default.properties**, in that order

Mutually Exclusive Arguments:

- b arg** Configure batch code specific properties files. Maintains **job.<arg>.properties** files.
- e arg** Configure general submitter mode properties files. Supports **THIN** and **LOCAL** execution mode. Maintains **submitbatch.<arg>.properties** files.

- l arg** Configure threadpool label specific properties files. Maintains **threadpoolworker.<arg>.properties** files.
- c** Configure the cluster configuration. Shortcut name for **tangoso1-coherence-override.xml**
- s** Configure the base submitter configuration. Shortcut name for **submitbatch.properties**.
- w** Configure the base threadpoolworker configuration. Shortcut name for **threadpoolworker.properties**.
- file** The properties file to edit. This must be a valid batch properties or Coherence XML configuration override file. May be used instead of the shortcut forms above and must be the last argument if specified.

Exclusive Argument:

- A** Generate ALL the base configuration files and exit. Combines **-c**, **-w** and **-s** option. Option **-A** must be the only option if specified.

Examples:

```
bedit threadpoolworker.properties
bedit -t wka -c
bedit -c -l cache
```

Note: For more information about **bedit[.sh]** refer to the [Configuring Batch using bedit](#) section of this guide.

buildJavadocsIndex[.sh] - Builds Appviewer java docs

Note: The **splenvirom[.sh]** utility must be executed before this utility can be used. See [splenvirom.sh](#) for details.

Note: This utility is only used for development environments where javadocs must be generated for other developers to use and appViewer is installed and deployed.

Note: This utility only works on custom java classes where source is provided with the customization installation and after the customizations have been installed.

The build AppViewer javadocs utility, **buildJavadocsIndex**, creates the javadocs for any custom java code for use for development. The utility does the following:

- After the custom java code has been installed using the SDK utilities, the utility generates the javadoc using the associated JDK.
- The utility optionally adds the generated to the javadoc to the product javadoc already installed in appViewer.
- Optionally regenerates the SPLWeb EAR file to update the appViewer.
- If running in embedded mode, the server needs to be restarted to reflect the change. If running in native mode the EAR file deployment needs to be updated.

Command Usage:

Linux/Unix:

```
buildJavadocsIndex.sh [-h] [-n]
```

Windows:

`buildJavadocsIndex.cmd [-h] [-n]`

Where:

- blank*** Generate javadocs and update SPLWeb EAR file
- h** Show usage.
- n** Generate javadocs but skip EAR update.

configureEnv – Setup Environment settings

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

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]) ([-c]|[-i]) [-h]`

Where:

- blank*** Configure basic configuration options
- a** Configure advanced configuration options
- c** Calculate variables only via silent mode. This refreshes ENVIRON.INI from individual changes
- 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.*

createDBStandalone[.sh] - Creates Database Patches

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See

[*splenviron.sh*](#) for details.

Note: This utility is used for patching only. Refer to the patch documentation for more information.

The **createDBStandalone[.sh]** utility, assembles all the needed components to allow database patches to be applied from a remote server. This utility creates the jar file named **db_patch_standalone.jar** containing the following:

- Relevant files in the **bin** directory needed for the patch.
- Relevant libraries needed for the patch.
- Relevant configuration files needed for the patch.

Command Usage:

Linux/Unix:

createDBStandalone.sh [-h] [-l <location>]

Windows:

createDBStandalone.cmd [-h] [-l <location>]

Where:

| | |
|----------------------------|--|
| blank | Performs complete extraction in default location SPLEBASE/tools/dbstandalone . |
| -h | Show usage. |
| -l <location> | Outputs jar file to the location <location> . If you do not specify a location then it will default to the <i>blank</i> action. |

extractCMSourceUNIX[.sh] - Migrate Code Across Upgrades

*Note: The [**splenviron\[.sh\]**](#) utility must be executed before this utility can be used. See [**splenviron.sh**](#) for details.*

Note: This utility is only used for extracting development source from one environment to another. This is used for initial migration from Oracle Utilities Application Framework V2.x to Oracle Utilities Application Framework V4.x

Note: This utility only works on custom java classes where source is provided with the customization installation and after the customizations have been installed.

Note: This utility is only provided on Linux/UNIX platforms.

When migrating from Oracle Utilities Application Framework Version 2.x to Oracle Utilities Application Framework Version 4.x, this utility extracts the source that was retained in the upgrade and reorganizes it to conform to the newer structures used in Oracle Utilities Application Framework Version 4.x. This is used as an initial load only and does not need to be executed again once the reorganization is complete.

The utility does the following:

- Existing code customizations are extracted to a central location.
- The code customization are reorganized into the Oracle Utilities Application Framework Version 4.x format.

The **applyCM** utility provided with the Oracle Utilities SDK can then be used to install the customizations on an initial environment as outlined in the [Software Configuration Management Series](#) (Doc Id: [560401.1](#)) available from [My Oracle Support](#).

Command Usage:

Linux/Unix:

`extractCMSourceUNIX.sh -v version -t target [-e] [-l lang] [-h]`

Where:

- h** Show usage.
- v *version*** Version identifier for custom build. This is the build number that is used for **CMVERSION.TXT**. This parameter is required.
- t *target*** Directory location for files to be housed. **applyCM** will be run against this location in future steps. This parameter is required.
- e** Include any custom AppViewer and/or Help content. This parameter is optional.
- l *lang*** Specific language code to extract for Help content. If blank, all languages are extracted. This parameter is optional.

genappvieweritems – generate AppViewer

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

*Note: This utility should only be executed if AppViewer is used in your environment (if the value of **Deploy Application Viewer Module** is set to **true**).*

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 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** 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.sh** 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.
- For customers using *native* mode, manual redeployment is necessary.
- Build/Rebuild the keystore files.

This concept is shown in the diagram below:

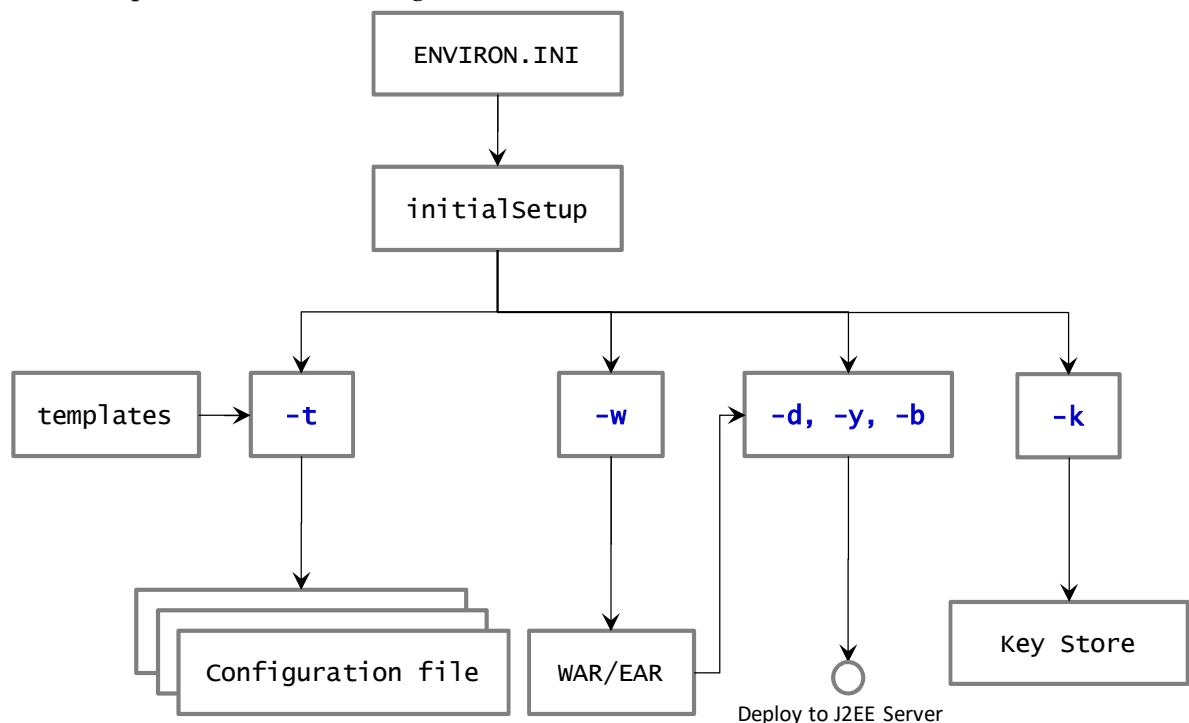


Figure 8 – initialSetup command structure

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 [ejb-jar.xml - Enterprise Java Server](#) Beans Configuration

The Business Application Server and MDB functionality uses Enterprise Java Server beans to implement service calls. By default the behavior of these components is defined and controlled with the `ejb-jar.xml` configuration file.

This configuration file is generated by the initialSetup using the `templates/ejb-jar.xml.template` file.

Note: This configuration file is typically NOT altered unless custom JMS or MDB resources are to be included in the servers within the product. Typically JMS resources are not included in the server Refer to the Oracle WebLogic JMS Integration (Doc Id: 1308181.1) whitepaper for details of JMS or MDB

advanced configuration.

ENVIRON.INI - Environment Configuration File.

Command Usage:

Linux/Unix:

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

Windows:

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

Where:

- blank** Process Templates, Build WAR/EAR files and Deploy to JEE Web Application Server in one process.
- h** Show usage.
- t** Process Templates only
- w** Build WAR/EAR files only
- v** Display Version
- s** Import Key Store from external location (**KS_IMPORT_KEYSTORE_FOLDER**)
- k** Generate Key store
- f** Force Task, skipping any warnings (used for silent installs only)

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
/sp1/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.*

installSFgroup[.sh] - Install Software Patches

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

*Note: This utility is located in the **tools/bin** directory of the software installation.*

Note: This utility does not handle prerequisites unless used with the [PatchDeployutility](#).

Note: This utility only does software installation. Database installation is handled by [ouafDatabasePatch](#).

The **installSFgroup** utility installs a single or group of software fixes that have been downloaded.

Command Usage:

Linux/Unix:

```
installSFgroup.sh -f InputFile -d InputDir -p product -v version -e splenviron -r splebase -l logfile
```

Windows:

```
installSFgroup -f InputFile -d InputDir -p product -v version -e splenviron -r splebase -l logfile
```

Where:

- f *InputFile*** Fill path of list of single fixes to install. This can be generated from [PatchDeployUtility](#), included in a service pack, included in a fix or manually created.
- d *InputDir*** Location of single fixes downloaded from My Oracle Support. These must be unzipped.
- p *product*** Product Name for patch file (this is the prefix of the file) (optional)
- v *version*** Version number of files (optional)
- e *splenviron*** Environment Name to apply fixes to. Defaults to current **SPLENVIRON** environment variable if not specified.
- r *splebase*** Root location of software for environment. Must match value in [/etc/cistab](#). Defaults to current **SPLEBASE** environment variable if not specified.
- l *logfile*** Name of log file to output. Defaults to **install.log** if not specified.

invokeDBUpdatePatch[.sh] - Regenerates passwords in Installation Options

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

The **invokeDBUpdatePatch[.sh]** utility, updates the Installation options that use encryption for key updates.

It is recommended to refer to the [Keystore Configuration](#) (Doc ID [2014161.1](#)) whitepaper for details of how and when to use this utility.

iwsdeploy - Inbound Web Services Deployment

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

The Inbound Web Services deployment utility, **iwsdeploy**, creates the Web Services EAR file and also deploys that EAR 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 EAR file called *Webservices.ear* located in the **\$SPLEBASE/splapp/iws/gen** directory (or **%SPLEBASE%\splapp\iws\gen**

directory on Windows).

- Deploys the EAR 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

jmxbatchclient[.sh] – JMX batch command line BATCH

*Note: The **splenvirom[.sh]** utility must be executed before this utility can be used. See [splenvirom.sh](#) for details.*

Note: For more examples of usage of this command refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) available from [My Oracle Support](#).

*Note: This facility will only operate if the **BATCH_RMI_PORT** is populated.*

The batch architecture While the JMX client interface provided allows real time information to be displayed in a JMX *browser*, if a JMX *browser* is not used then the JMX interface may be interfaced using a command line utility. This utility is useful to allow third party products (such as batch schedulers) or other systems to control and monitor the state of the system.

This JMX batch command line allows the following to be performed:

- Identify what thread pools are defined in a **threadpoolworker**
- See what active batch processes or threads are currently running
- Be able to cancel a particular thread or a batch process
- Gracefully shutdown a **threadpoolworker**
- The command line utility is in the following format:

To execute the command line, the administrator must:

- Logon to the machine running the product (any tier where the product software exists).
- Attach to the environment using the **splenvirom[.sh]** command. This sets the appropriate environment settings for the script.
- Execute the JMX Batch command line utility:

jmxbatchclient[.sh] -j [URL] [options]

Where **[options]** are:

- c Specifies that active threads should be cancelled. Can be used with **-f** option to cancel only batch processes matching the regular expression provided. For example:

Note: Cancelled threads are marked with the date, time, userid and IP address of the user who initiated the cancel command.

- d Display the details of the currently active threads.

- f If a large number of threads are currently active, a filter can be supplied to only display or cancel threads that match the *regex* based pattern.
For example the threadpool be filtered to show only the BAT1 with the option: -f *.*BAT1.** as follows:

```
jmxbatchclient.sh -j
service:jmx:rmi:///jndi/rmi://myserver:9999/oracle/ouaf/batchConnector -f .*BAT1.*
```

would yield:

```
Options: -j
service:jmx:rmi:///jndi/rmi://myserver:9999/oracle/ouaf/batchConnector -f .*BAT1.*
Connecting to
service:jmx:rmi:///jndi/rmi://myserver:9999/oracle/ouaf/batchConnector
ActiveGridNode
threadPools=[MYSERVER:5,
LOCAL_THREAD_POOL:b9835d11f15fd71b:681ba91d:1200151a3c8:-8000:0,
SCHEDULER_DAEMON_THREAD_POOL:1]
BatchThread_ZZQABAT1_1_of_1.31
```
- h Display the available options and their descriptions.
- j JMX URL to perform the action against (Required).
This should match the `spl.runtime.management.connector.url.default` property specified in the `threadpoolworker.properties`.
- k Specifying this option will result in the cancellation of all currently running threads and the stoppage of the threadpoolworker process.

*Note: Active threads within a cancelled **threadpoolworker** are marked with the date, time, userid and IP address of the user who initiated the kill command*

- l By default, all logging information is displayed and logged using log4j2. Supplying this option will result in only select information being displayed to the system output.
- s Display the summary of the currently active threads is a listing format.
- t Amount of time, in ms, to wait as expressed in `<wait>` for active jobs to cancel.
`<wait>`
>
- z Display the cache details. Must be used in conjunction with -j option.

ouafDatabasePatch[.sh] - Install Database Patches

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: For more information about patching refer to the [Software Configuration Management Series](#) (Doc Id: [560401.1](#)) available from [My Oracle Support](#).

The **ouafDatabasePatch** allows database patches to be installed from the command line or

via the Application Management Pack for Oracle Utilities. This utility is a companion to with [installSFgroup](#) which installs the software component of patches.

Command Usage:

Linux/Unix:

```
ouafDatabasePatch.sh [-h] [-u] [-v] [-x] [-t toolsdir] [-p ouafparms]
```

Windows:

```
ouafDatabasePatch [-h] [-u] [-v] [-x] [-t toolsdir] [-p ouafparms]
```

Where:

- | | |
|-----------|--|
| -h | Show usage for patch. |
| -u | Show usage of command |
| -v | Display version of patch |
| -x | Database Password for applying patch |
| -b | Location of tools bin directory |
| -p | Additional parameters passed to utility (as per patch documentation). This must be the last parameter on the command line. This is primarily used by the Application Management Pack for Oracle Utilities. |

PatchDeployUtility[.sh] - Create and Deploy Patch Rollups

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: For more information about patching refer to the [Software Configuration Management Series](#) (Doc Id: [560401.1](#)) available from [My Oracle Support](#).

The **PatchDeployUtility** allows prerequisite product software and database patches to be analyzed, downloaded and applied in a single installation. This utility can be used from the command line or via the Application Management Pack for Oracle Utilities.

There are three phases for the utility to execute:

- **Report** Mode - Generate a patch dependency list for a particular patch.
- **Download** Mode - Download the patches not already downloaded from the Report mode list.
- **Rollup** Mode - Create a single installer to install all the patches in the patch dependency list in one install.

Note: This utility does not install the patches. They must be installed using the [installSFgroup](#) and [ouafDatabasePatch](#) utilities.

Command Usage:

Linux/Unix:

```
PatchDeployUtility.sh -f patchfile -m mode -l missingfile -d dir -r dir -n missingfile
```

Windows:

```
PatchDeployUtility.sh -f patchfile -m mode -l missingfile -d dir -r dir -n missingfile
```

Where:

- f *patchfile*** The full path and file name to the **patchdependency.txt** file of the patch to use as a reference point. This parameter is used by the **REPORT** mode only and is required for that mode.
- m *mode*** Mode. Supported values: **REPORT**, **DOWNLOAD** or **ROLLUP**.
- l *missingfile*** Full path and file name for output for **REPORT** mode. This file will contain the list of fixes that have not been downloaded. This parameter is used by the **REPORT** mode only and is required for that mode.
- d *dir*** Full path to location where any missing patches will be downloaded to when using the **DOWNLOAD** mode. This is also used by the **ROLLUP** mode as a source directory.
- r *dir*** Fill to location where the rollup files will be generated. This is the location used by the **installSFGGroup** and **ouafDatabasePatch** utilities will be executed from. This parameter is by the **ROLLUP** mode only and is required for that mode.
- n *missingfile*** Full patch and file name of the missing patches generated by the **REPORT** mode. This parameter is used by **DOWNLOAD** and **ROLLUP** modes.

For example:

```
PatchDeployUtility.sh -f
/scratch/mypatches/fix10001/data/etc/patchdependency.txt -m REPORT -l
/scratch/patches/missing.txt
PatchDeployUtility.sh -n /scratch/patches/missing.txt -m DOWNLOAD -d
/scratch/mypatches
PatchDeployUtility.sh -n /scratch/patches/missing.txt -m ROLLUP -d
/scratch/mypatches -r /scratch/myrollups
```

showjavaproc[.sh] - Show java processes

*Note: The **splenviron[.sh]** utility must be executed before this utility can be used. See [splenviron.sh](#) for details.*

Note: This utility is only provided on Linux/UNIX platforms.

Note: This utility is mainly useful for embedded installations. It can be used for native installations but the environment name will not be rendered.

The **showjavaproc** utility lists the process ids for the WebLogic servers installed on the machine.

Command Usage:

Linux/Unix:

```
showjavaproc.sh [-e envname] [-q]
```

Where:

- blank** Find all of the WebLogic servers running on the machine
- e *envname*** Filter the output by environment name. If it is found then output a 1 return code
- q** Do not put output on screen. Useful for custom scripting.

For example:

```
showjavaproc.sh -e MYENVIRON || echo "Weblogic Java Process is running"
showjavaproc.sh -e MYENVIRON && echo "Weblogic Java Process is NOT running"
```

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.

starttpw[.sh] - Start Threadpoolworker

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: This utility is only provided on Linux/UNIX platforms.

Note: This utility is useful for developers.

This utility is a wrapper utility that calls the threadpoolworker script for use in command line batch submitters or from the command line.

Command Usage:

Linux/Unix:

```
starttpw.sh [-h] [-Q] [threadpoolworker options]
```

Where:

- Blank** Execute the [\[threadpoolworker options\]](#) in the background (nohup).
- h** Show usage.
- Q** Execute the [\[threadpoolworker options\]](#) in the foreground

submitjob[.sh] - Submit Batch Threads **BATCH**

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: For more examples of usage of this command refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) available from [My Oracle Support](#).

The following options can be specified when executing utility submitjob[.sh]:

```
submitjob[.sh] [-B][-b][-c][-d][-e][-f][-g][-h][-i][-j][-l][-L][-m] [-n] [-p] [-P] [-Q] [-R] [-r] [-s] [-t] [-u] [-x] [-X]
```

where command line options are:

- b <batch code>** Batch code of the batch process to submit. When submitting a batch process, a batch code is always required. Either this option or **-P** may be specified, not both.
If this option is specified, **submitjob** will use the supplied batch code to look for a default properties file for that batch code (e.g. **VAL-SA.properties** or **VAL-SA.properties.xml** as discussed above) and use those properties if found.
- c <thread count>** Concurrent number of threads in which to run the process.
- d <date>** Process / business date.
Format is YYYY-MM-DD

| | |
|---|---|
| <code>-e < THIN CLUSTERED></code> | <p>Execution mode for this submission. If execution mode THIN is selected, the JVM will create a full application context and run the batch process inside the JVM – i.e. it will not be submitted to a thread pool for a worker JVM to pick up and run. This is analogous to running the batch process using the existing SPLBATCH[.sh] utility.</p> <p>THIN mode is for development purposes only.</p> <p>If CLUSTERED is selected, the batch process will be submitted to run in the specified distributed thread pool (option -p). It is also possible to have the submitter JVM be a worker JVM and run the batch process (similar to THIN mode, but in parallel threads). See option -L.</p> |
| <code>-f <record count></code> | Record commit frequency count. |
| <code>-g <four Y N switches></code> | <p>Positional tracing switches:</p> <ol style="list-style-type: none"> 1. Program entry 2. Program exit 3. SQL statements 4. General program debugging info <p>For example, NNYN will trace all SQL statements. Value of NNNN disables all tracing.</p> |
| <code>-h</code> | Show help information. Display the available options and their descriptions. The information is not logged. |
| <code>-i <RMI port number></code> | Port number of RMI Registry to start and/or reference. If specified with -R , this number will be used only to substitute applicable URL {port} references. This option will not add any new RMI/JMX properties - it can only be used to override existing ones. <i>See note below</i> |
| <code>-J</code> | <p>Do not start JMX connector. This option disables JMX monitoring for this JVM. As far as submitjob is concerned, options -i, -R and -J are only applicable to batch processes submitted in THIN mode, or CLUSTERED mode to the LOCAL thread pool.</p> <p>For each property prefixed by spl.runtime.management.connector.url that is defined with the default set of properties (e.g. in the submitbatch.properties file), the framework will start a JMX Connector for the specified URL.</p> |

- This activates JMX monitoring inside the worker node so that a client JMX console can be used to monitor and manage active threads. If this option is specified, the framework will not start any JMX connectors.
- l** <ENG|FRA|etc.> Language code. Relevant language pack must be installed.
- L** Submit this batch to the LOCAL thread pool (i.e. this JVM). Only applicable for **CLUSTERED** mode. If specified, any default thread pool property is ignored. This option and **-p** are mutually exclusive.
- By specifying option **-L**, the batch process is submitted to the LOCAL thread pool that every submitter JVM offers by default. This option is only applicable in a **CLUSTERED** mode execution (**-e**). This is similar to submitting the batch process in **THIN** mode (i.e. a worker JVM is not needed to run the batch process), except thread pool LOCAL can run multiple batch threads concurrently.
- For example, the following command will run batch process VAL-SA inside this submitter JVM (LOCAL thread pool) in 8 threads concurrently:
submitjob[.sh] -b VAL-SA -c 8 -L -e CLUSTERED
- n** <email address> Send a notification email when a batch process has ended to <email address>. See [Sending emails at the conclusion of batch process](#) for more information.
- p** <threadpool name> Distributed thread pool in which to run the batch process. This option and **-L** are mutually exclusive.
- P** Issue console prompts for the standard batch process parameters. When submitting a batch process, a batch code is always required. Either this option or **-b** may be specified, not both. If **-P** is specified, the submitter JVM will prompt for the batch code and other run parameters. If a batch-specific properties file exists for the batch code entered at the prompt, it will NOT be used; the only defaults in effect would be the ones specified in **submitbatch.properties**.
- Q** Preview the properties that would be in use for

| | |
|---|---|
| | the run without actually running the application. Specify other options along with this option to show how they would override or substitute the default properties. The information is not logged. |
| -R | Do not start a local RMI registry. As far as submitjob is concerned, options -i , -R and -J are only applicable to batch processes submitted in THIN mode, or CLUSTERED mode to the LOCAL thread pool. If property spl.runtime.management.rmi.port is defined as a default property (e.g. in the submitbatch.properties file), the batch framework will attempt to start an RMI registry on the given port number. This option can be used to suppress the automatic RMI registry startup. It may be required if an externally started RMI registry is already running. Note that if this option is used, the RMI port number supplied through the -i option is only used for substitution in the JMX Connector URLs. |
| -r <run number> | Run number of batch process to rerun. |
| -s <space name> | Space name for <i>hard partition</i> of workers. Default is MAIN . Used for development only. |
| -t <thread number> | Number of individual thread for this submission. Specify 0 to automatically submit all threads. |
| -u <user id> | Application user id used for batch process |
| -x <name=value,name=value,...> | Name=value pairs of INDIVIDUAL soft parameters expected by the batch program. Value portion may be enclosed in quotes. These parameters will be merged with any existing (defaulted) soft parameters. This option and -X are mutually exclusive. |
| -X <name=value,name=value,...> | Name=value pairs of ALL soft parameters expected by the batch program. Value portion may be enclosed in quotes. These parameters will replace all existing (defaulted) soft parameters. This option and -x are mutually exclusive. |
| -z <jvmoptions> | Java options to use for submitter. |

This utility returns two return codes:

- Zero (0) - The job or thread was successful.
- Non-Zero - The job or thread was NOT successful.

threadpoolworker[.sh] - Manage Threadpools **BATCH**

Note: The [splenviron\[.sh\]](#) utility must be executed before this utility can be used. See [splenviron.sh](#) for details.

Note: For more examples of usage of this command refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) available from [My Oracle Support](#).

The following options can be specified when executing script threadpoolworker.

`threadpoolworker[.sh] [-d] [-e] [-h] [-i] [-j] [-p] [-q] [-r] [-s]`

Where command line options are:

- | | |
|--|--|
| <code>-d <Y N></code> | Whether the node is acting as a scheduler daemon. Specify N for No and Y for Yes. If you are already using a scheduler daemon in the online system or are not using online submission then set this to N . Default is N . |
| <code>-e < THIN CLUSTERED></code> | Execution mode for this threadpool. If CLUSTERED is the threadpool will join the cluster specified in the threadpoolworker.properties file. |
| <code>-h</code> | Show command line help. List the available options and their descriptions. It is formatted for a 121-column width display. The information is not logged. |
| <code>-i <RMI Port></code> | Override port number for JMX. If specified with <code>-R</code> , this number will be used only to substitute applicable URL {port} references. This option will not add any new RMI/JMX properties - it can only be used to override existing ones. This option specifies the port number to: <ul style="list-style-type: none"> • Use when the framework starts an RMI Registry and • Substitute in all JMX Connector URL {port} references. |
| <code>-j</code> | Do not start JMX monitoring. For each property prefixed by spl.runtime.management.connector.url that is defined with the default set of properties (e.g. in the threadpoolworker.properties file), the framework will start a JMX Connector for the specified URL. This activates JMX monitoring inside the worker node so that a client JMX console can be used to monitor and manage active threads. If this option is specified, the framework will not start any JMX connectors. |
| <code>-l <READ_ONLY READ_WRITE OFF></code> | Enable or disable batch caching. Default: READ_WRITE . OFF and READ_ONLY are reserved for |

| | |
|---------------------------------------|--|
| | specific processes |
| -p <name=value,name=value,...> | Thread pool(s) offered by this worker node. Consists of one or more name=value pairs, where "name" is the name of the pool and "value" the number of threads offered in the pool. For example, DEFAULT=5,ONLINE=3 |
| -Q | Preview the properties that would be active for this run. Used for testing. Preview the properties that would be in use for the run without actually running the application. Specify other options along with this option to show how they would merge with, override or substitute the default properties. The information is not logged. |
| -R | Do not start a local RMI registry. If property spl.runtime.management.rmi.port is defined as a default property (e.g. in the threadpoolworker.properties file), the batch framework will attempt to start an RMI registry on the given port number. This option can be used to suppress the automatic RMI registry startup. It may be required if an externally started RMI registry is already running. |
| | <i>Note: If this option is used, the RMI port number supplied through the -i option is only used for substitution in the JMX Connector URLs.</i> |
| -s <space name> | Space name for "hard partition" of workers. Default is MAIN. Reserved for internal use only. |
| -sf <file-path> | Generate or append command to a script. Reserved for system use. |
| -sl <line to add to script> | Add line to script reference. Reserved for system use. |
| -z <jvm options> | Add or override JVM options for threadpool. |

When **threadpoolworker** is invoked, the command-line options will alter its default configuration. The default configuration options come from either internal system defaults or the **threadpoolworker.properties** file described above.

The properties are overridden in the following order:

1. The **threadpoolworker.properties** supersedes the internal system defaults.
2. The command-line options supersede the defaults in **threadpoolworker.properties** and the internal system defaults.

Example 1

Assuming we have the above set of properties in **threadpoolworker.properties** and script **threadpoolworker** is invoked as follows:

```
threadpoolworker[.sh] -d Y
```

This will replace the default "daemon" property to "N" (i.e. false) so that the properties now look as follows:

```
com.splwg.grid.distThreadPool.threads.DEFAULT=5
com.splwg.grid.distThreadPool.threads.LOCAL=0
com.splwg.batch.scheduler.daemon=false
spl.runtime.management.rmi.port=9999
spl.runtime.management.connector.url.default=service:jmx:rmi:///jndi/rmi://{host}:{port}/oracle/ouaf/batchConnector
```

Starting and Stopping Using consoles

In the native mode installation there are a number of ways to start and stop the product:

- Logon to the Oracle WebLogic Administration console for the domain and navigate to the servers section where the product is installed. Using the Control function, start or stop the servers as necessary.

Note: This facility also allows enabling or disabling Administration Mode for a server. This allows administration tasks on that server while disabling end user access.

- For customers using Oracle Enterprise Manager, logon to the Oracle Enterprise Manager console the product can be started and stopped using the following target types:
 - WebLogic Domain
 - WebLogic Server for product servers
 - Domain Application Deployment

Note: Refer to [Oracle Weblogic Console Operations](#) section for starting and stopping information.

- For customers using the Oracle Application Management Pack for Oracle Utilities¹¹ the product can be started and stopped using the following target types:
 - Oracle Utilities Environment - Starts and Stops all child targets in a single operation.
 - Oracle Utilities Web Application
 - Oracle Utilities Web Services
 - Oracle Utilities Batch Server

Note: Refer to [Operations from Oracle Enterprise Manager](#) section for information on operations from OEM.

Monitoring

This section outlines some basic monitoring regimes and methods for the product. It is highly recommended that you read the [Performance Troubleshooting Guides](#) (Doc 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 identify 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 *tracert* 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 tracert 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 some 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-user.log** and **spl_service-user.log** but **spl_service-user.log** may or not appear depending on the installation type, therefore the name **spl*.log** is used.*

All the logs displayed under this feature pertain to business/user logs. No technical logs would be displayed.

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.
- **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 toolbar on the browser screen as shown below:



Figure 9 – Debug toolbar

- **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.
- **JS Console** - enable JavaScript console
- 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:

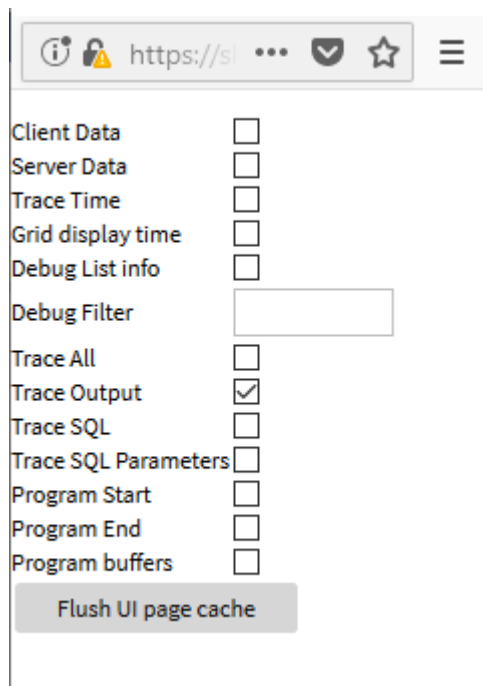


Figure 10 – Debug controls

- 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 JEE 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 7 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 7 (and above).
- **Oracle Enterprise Manager** - Customers using Oracle WebLogic can use the Fusion Middleware targets available within Oracle Enterprise Manager to monitor online and web services applications. Customers using Application Management Pack for Oracle Utilities can use monitoring capabilities of those targets as well.
- **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 [Performance Troubleshooting Guides](#) (Doc 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 JEE 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 displayed 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 |
|--|---|
| <code>WEB_JMX_RMI_PORT_PERFORMANCE</code> | Port Number used for JMX based management for Web Application Server. |
| <code>ouaf.jmx.splwls.g.base.support.management.mbean.JVMInfo</code> | Globally enable or disable JVMInfo Mbean (setting in spl.properties). Default is enabled . |
| <code>ouaf.jmx.com.splwg.base.web.mbeans.FlushBean</code> | Globally enable or disable FlushBean Mbean (setting in spl.properties). Default is enabled . |
| <code>ouaf.jmx.com.oracle.ouaf.uws.mbeans.WSFlushBean</code> | Globally enable or disable WSFlushBean Mbean (setting in spl.properties). Default is enabled . |
| <code>BSN_JMX_RMI_PORT_PERFORMANCE</code> | Port Number used for JMX based management for Business Application Server. |
| <code>ouaf.jmx.com.splwg.ejb.service.management.PerformanceStatistics</code> | Globally enable or disable PerformanceStatistics Mbean (setting in spl.properties). Default is enabled . |
| <code>BSN_JMX_SYSUSER</code> | Default JMX Userid for both Web Application Server and Business Application Server |
| <code>BSN_JMX_SYSPASS</code> | Default JMX Password for both Web Application Server and Business Application Server |
| <code>BATCH_RMI_PORT</code> BATCH | Port Number used for JMX based management for background processing. |

| Configuration Setting | Deployment details |
|---|---|
| <code>WEB_BATCH_CLUSTER_URL</code> BATCH | Cluster URL override to be used for JMX based management for background processing. This can manually be set to identify an administration node for processing. |

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`

and

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

Where:

- `<host>` The Web Application Server host name
- `<jmx_port>` The JMX Port specified using **WEB_JMX_RMI_PORT_PERFORMANCE** for the online and **WEB_JMX_RMI_PORT_PERFORMANCE** for Inbound Web Services 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*):

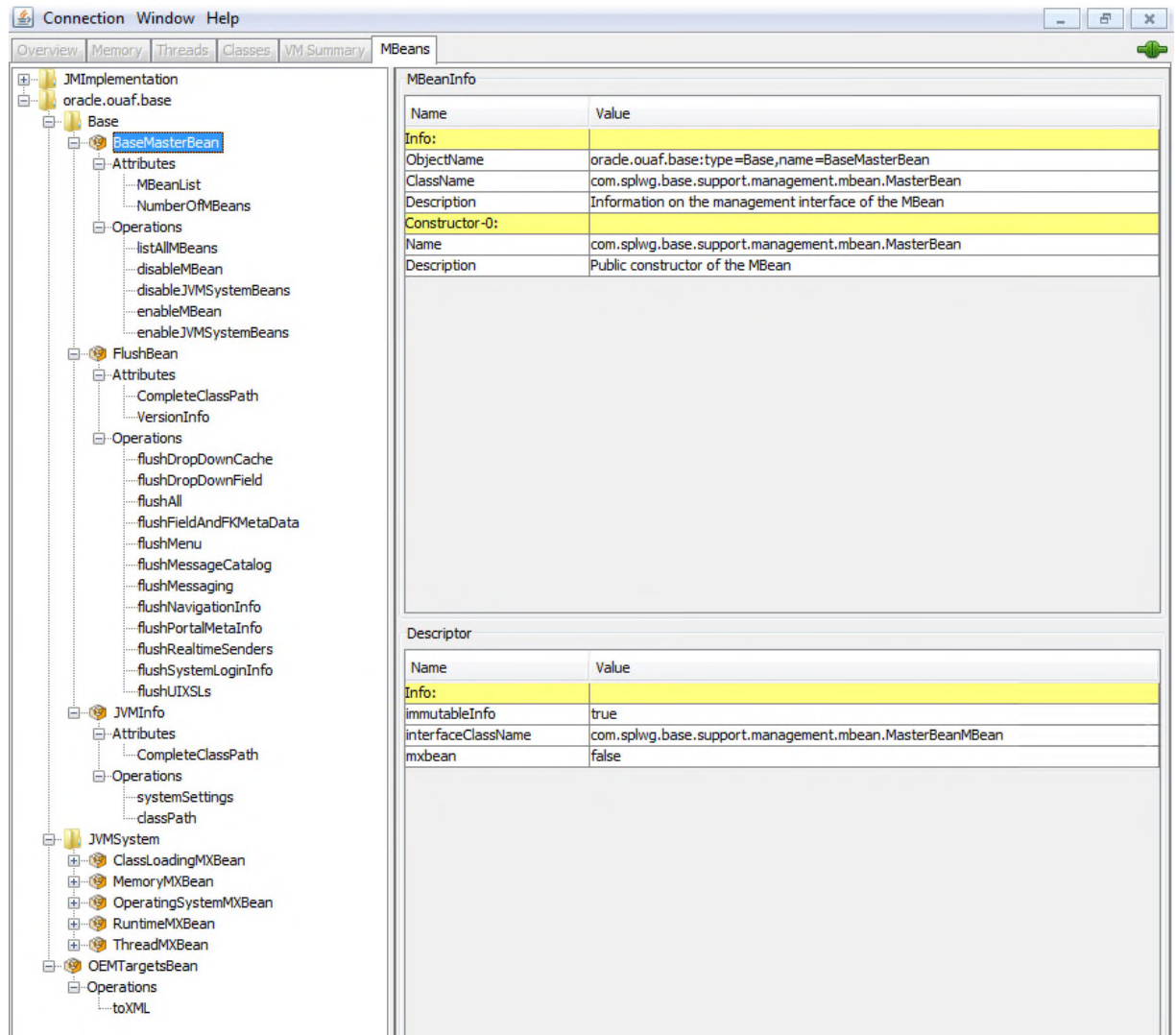


Figure 11 – Example Web Application Server Mbeans in jconsole

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

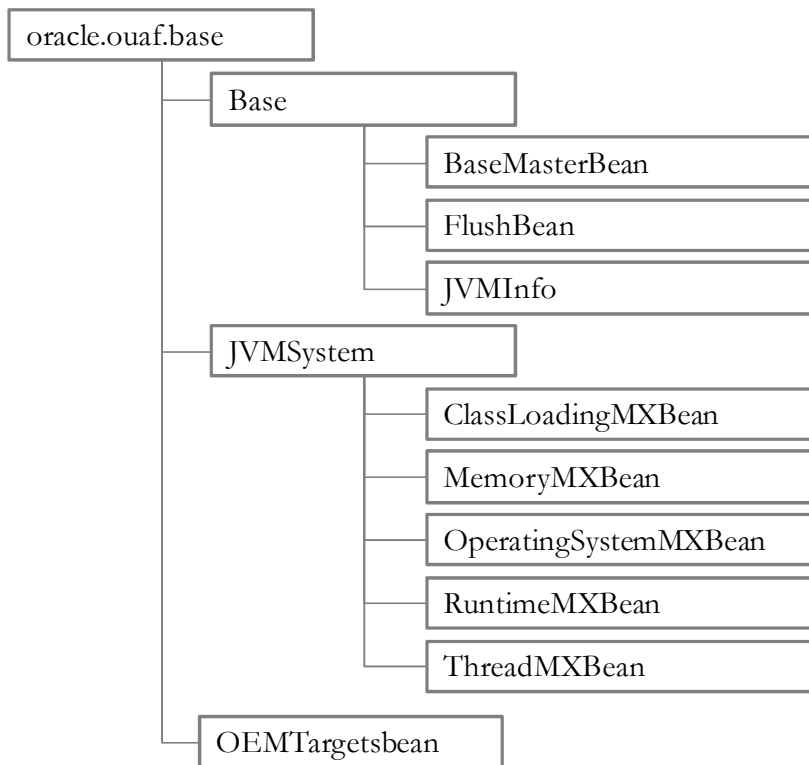


Figure 12 – Web Application Server Mbeans

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

| Mbean | Usage |
|------------------------------|--|
| BaseMasterBean | Master MBean |
| Attributes | |
| NumberOfMbeans | Returns number of active Mbeans |
| MBeanList | Returns an array with the list of Mbeans defined to this Master Mbean |
| Operations | |
| disableMbean | Disables Mbean with designated name as a par |
| enableMbean | Enables Mbean with designated name |
| disableMbean | 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 | Enables base JVM system Mbeans |
| disableJVMSystemBeans | disables base JVM system Mbeans |
| FlushBean/WSFlushBean | Manages Online/IWS Cache |
| Attributes | |
| VersionInfo | Returns string of base version number of Mbean. |

| Mbean | Usage |
|--------------------------------|---|
| completeClassPath | Returns classpath name of Mbean. |
| Operations | |
| flushAll | Reset all elements in online data cache |
| flushDropDownCache | Resets cached elements of the online drop down lists in online data cache |
| flushDropDownField | Resets drop down fields in online data cache (<i>Development use only</i>) |
| flushFieldAndFKMetaData | Resets Field and Foreign Key Meta Data in online data cache |
| flushMenu | Reset Menu items in online data cache |
| flushMessageCatalog | Reset field labels in online data cache |
| flushMessaging | Reset messages in online data cache |
| flushNavigationInfo | Reset navigation information in online data cache |
| flushPortalMetaInfo | Reset portal and zone information in online data cache |
| flushSystemLoginInfo | Reset security information in online data cache |
| flushUIXSLS | Reset user interface style sheets in online data cache |
| JVMInfo | JVM Information Mbean |
| Attributes | |
| completeClassPath | Displays the class path of the JVMInfo mbean |
| Operations | |
| classpath | Returns the full classpath used by the online JVM |
| systemSettings | Returns the attributes of the JVM for debugging and support purposes. |
| JVMSystem | JVM Mbean (Java base metrics) |
| ClassLoadingMXBean | Class Loading JVM Mbean |
| Attributes | |
| LoadedClassCount | Returns the number of classes that are currently loaded in the JVM |
| ObjectName | The ObjectName for uniquely identifying the MXBean for the class loading system within an MBeanServer. In this case it is set to java.lang:type=ClassLoading |
| TotalLoadedClassCount | Returns the total number of classes that have been loaded since the JVM was last started. |
| UnloadedClassCount | Returns the total number of classes unloaded since the |

| Mbean | Usage |
|--------------------------------------|--|
| | Java virtual machine has started execution. |
| Verbose | Enables or disables the verbose output for the class loading system. Default is false (<i>disabled</i>) |
| MemoryMXBean | Memory JVM Mbean |
| Attributes | |
| HeapMemoryUsage | 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 | Returns the current memory usage of non-heap memory that is used by the JVM. Initial, Committed, Maximum and Used memory statistics are provided for Non-Heap memory |
| ObjectPendingFinalization | Returns the approximate number of objects for which finalization is pending (used for diagnosing memory leaks). |
| ObjectName | The ObjectName for uniquely identifying the MXBean for the memory system within an MBeanServer. In this case it is set to java.lang:type=Memory |
| Verbose | Enables or disables the verbose output for the memory system. Default is false (<i>disabled</i>) |
| Operations | |
| gc | Initiate garbage collection |
| Notifications | |
| javax.management.Notification | Used for low memory notifications. Notification Types supported: (java.management.memory.threshold.exceeded, java.management.memory.collection.threshold.exceeded) |
| OperatingSystemMXBean | Operating System Mbean |
| Attributes | |
| MaxFileDescriptorCount | Returns the File Descriptor Maximum Limit in force on the JVM |
| OpenFileDescriptorCount | Returns the number of Open File Descriptors currently used by JVM |
| CommittedVirtualMemorySize | Returns the amount of committed virtual memory (that is, the amount of virtual memory guaranteed to be available to the running process). |

| Mbean | Usage |
|--------------------------------|--|
| FreePhysicalMemorySize | Returns the total amount of free physical memory |
| FreeSwapSpaceSize | Returns the total amount of free swap space |
| ProcessCpuTime | Returns the amount of process CPU time consumed by the JVM |
| TotalPhysicalMemorySize | Returns the total amount of physical memory |
| TotalSwapSpaceSize | Returns the total amount of swap space |
| Name | Returns the operating system name |
| Version | Returns the version of the operating system |
| Arch | Returns the operating system architecture |
| AvailableProcessors | Returns the number of available processors to the JVM |
| SystemLoadAverage | Returns the system load average for the last minute. |
| Name | 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 | Returns the Java class path that is used by the system class loader to search for class files. |
| ObjectName | The ObjectName for uniquely identifying the MXBean for the operating system within an MBeanServer. In this case it is set to java.lang:type=OperatingSystem |
| RuntimeMXBean | Java Runtime MBean |
| Attributes | |
| Starttime | Returns the start time of the Java virtual machine in milliseconds. This method returns the approximate time when the JVM started. |
| ManagementSpecVersion | Returns the version of the specification for the management interface implemented by the running JVM |
| VmName | Returns the Java virtual machine implementation name |
| VmVendor | Returns the Java virtual machine implementation vendor |
| VmVersion | Returns the Java virtual machine implementation version |
| SpecName | Returns the Java virtual machine specification name |
| SpecVendor | Returns the Java virtual machine specification vendor |

| Mbean | Usage |
|--|--|
| SpecVersion | Returns the Java virtual machine specification version |
| LibraryPath | Returns the Java library path |
| BootClassPath | Returns the boot class path that is used by the bootstrap class loader to search for class files |
| Uptime | Returns the uptime of the Java virtual machine in milliseconds |
| BootClassPathSupported | Tests if the JVM supports the boot class path mechanism used by the bootstrap class loader to search for class files. Returns <i>false</i> if not supported; <i>true</i> if supported |
| InputArguments | 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 | Returns a map of names and values of all system properties |
| ThreadCount | Returns the current number of live threads including both daemon and non-daemon threads |
| PeakThreadCount | Returns the peak live thread count since the JVM started or peak was reset |
| TotalStartedThreadCount | Returns the total number of threads created and also started since the JVM started |
| DaemonThreadCount | Returns the current number of live daemon threads |
| ThreadContentionMonitoringSupported | Tests if the JVM supports thread contention monitoring. Returns <i>false</i> if not supported; <i>true</i> if supported |
| ThreadContentionMonitoringEnabled | Enables or disables thread contention monitoring. Set to <i>false</i> to disable; <i>true</i> to enable. |
| CurrentThreadCpuTime | 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 | 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 |

| Mbean | Usage |
|--------------------------------------|---|
| | nanoseconds accuracy. |
| ThreadCpuTimeSupported | Tests if the JVM supports CPU time measurement for the current thread. Returns <i>false</i> if not supported; <i>true</i> if supported |
| ThreadCpuTimeEnabled | Enables or disables thread CPU time measurement. The default is platform dependent. Set to <i>false</i> to disable; <i>true</i> to enable. |
| CurrentThreadCpuTimeSupported | Tests if the Java virtual machine supports CPU time measurement for the current thread. Returns <i>false</i> if not supported; <i>true</i> if supported |
| ObjectMonitorUsageSupported | Tests if the Java virtual machine supports monitoring of object monitor usage. Returns <i>false</i> if not supported; <i>true</i> if supported |
| SynchronizerUsageSupported | Tests if the JVM supports monitoring of ownable synchronizer usage. Returns <i>false</i> if not supported; <i>true</i> if supported |
| AllThreadIds | Returns all live thread IDs. Some threads included in the returned array may have been terminated when this method returns |
| ObjectName | The ObjectName for uniquely identifying the MXBean for the java runtime within an MBeanServer. In this case it is set to java.lang:type=Runtime |
| Operations | |
| dumpAllThreads | 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 <ul style="list-style-type: none"> • Locked Monitors - if <i>true</i>, dump all locked monitors • Locked Synchronizers - if <i>true</i>, dump all locked ownable synchronizers |
| findDeadlockedThreads | 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 | 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 |

| Mbean | Usage |
|-----------------------------------|--|
| | 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 |
| <code>getThreadInfo</code> | Returns the thread info for a thread of the specified id with no stack trace. |
| <code>getThreadInfo</code> | Returns the thread info for each thread whose ID is in the input array ids with no stack trace. |
| <code>getThreadInfo</code> | Returns thread information for a thread of the specified id, with stack trace of a specified number of stack trace elements. The <i>maxDepth</i> parameter indicates the maximum number of <i>StackTraceElements</i> to be retrieved from the stack trace. This method does not obtain the locked monitors and locked synchronizers of the thread |
| <code>getThreadInfo</code> | 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 <i>maxDepth</i> parameter indicates the maximum number of <i>StackTraceElements</i> to be retrieved from the stack trace. This method does not obtain the locked monitors and locked synchronizers of the threads |
| <code>getThreadInfo</code> | <p>Returns the thread info for each thread whose ID is in the input array ids, with stack trace and synchronization information.</p> <p>This operation obtains a snapshot of the thread information for each thread including:</p> <ul style="list-style-type: none"> • the entire stack trace, • the object monitors currently locked by the thread if <i>lockedMonitors</i> is true, and • the ownable synchronizers currently locked by the thread if <i>lockedSynchronizers</i> is true |
| <code>getThreadUserTime</code> | Returns the CPU time that a thread of the specified ID has executed in user mode in nanoseconds |
| <code>resetPeakThreadCount</code> | Resets the peak thread count to the current number of live threads |
| OEMTargetsBean | Application Management Pack for Oracle Utilities interface |
| Operations | |
| <code>toXML</code> | Generate Target information for Oracle Enterprise Manager. An XML document with information for the |

Mbean**Usage**

Application Management Pack for Oracle Utilities.

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 screenshot shows the jconsole interface with the MBeans tab selected. The left pane displays a tree view of the MBean hierarchy, with 'PerformanceStatistics' selected under 'Performance'. The right pane shows the 'MBeanInfo' details for the selected MBean.

| Name | Value |
|-----------------------|--|
| Info: | |
| ObjectName | oracle.ouaf.base:type=Performance,name=PerformanceStatistics |
| ClassName | com.splwg.ejb.service.management.PerformanceStatistics |
| Description | Information on the management interface of the MBean |
| Constructor-0: | |
| Name | com.splwg.ejb.service.management.PerformanceStatistics |
| Description | Public constructor of the MBean |

| Name | Value |
|--------------------|---|
| Info: | |
| immutableInfo | true |
| interfaceClassName | com.splwg.ejb.service.management.PerformanceStatisticsMBean |
| mxbean | false |

Figure 13 – Example Business Application Server Mbeans in jconsole

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

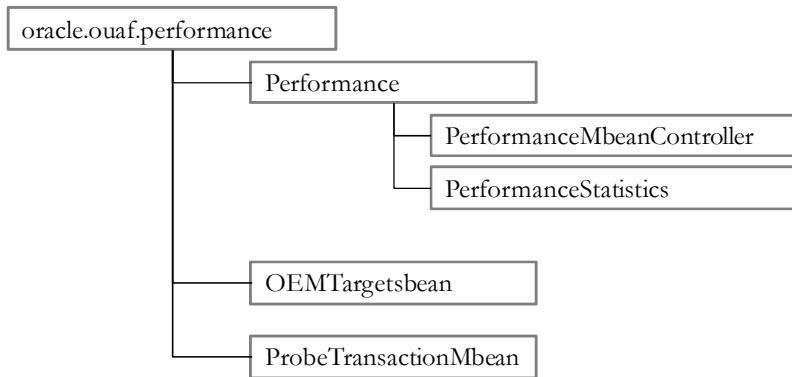


Figure 14 – Business Application Server Mbeans

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

| Mbean | Usage |
|-----------------------------------|---|
| PerformanceMbeanController | Performance Mbean Controller |
| Operations | |
| disableMbean | Disable PerformanceStatistics Mbean |
| enableMbean | Enable PerformanceStatistics Mbean |
| PerformanceStatistics | Performance Statistics |
| Attributes | |
| ReadCount | Returns number of executed <i>read object</i> calls since last reset or last time collection enabled |
| DeleteCount | Returns number of executed <i>delete object</i> calls since last reset or last time collection enabled |
| ChangeCount | Returns number of executed <i>change object</i> calls since last reset or last time collection enabled |
| AddCount | Returns number of executed <i>add object</i> calls since last reset or last time collection enabled |
| DefaultItemCount | Returns number of executed calls to <i>default the object values</i> since last reset or last time collection enabled |
| ExecuteBOCount | Returns number of calls to <i>Business Objects</i> since last reset or last time collection enabled |
| ExecuteBSCount | Returns number of calls to <i>Business Services</i> since last reset or last time collection enabled |
| ExecuteListCount | Returns number of calls to <i>List based services</i> since last reset or last time collection enabled |
| ExecuteSearchCount | Returns number of calls to <i>search based services</i> since last reset or last time collection enabled |
| ReadSystemCount | Returns number of calls to <i>Oracle Utilities Application Framework system Objects</i> since last reset or last time |

| Mbean | Usage |
|--------------------------------------|--|
| | collection enabled |
| ValidateCount | Returns number of calls to <i>Validate objects</i> since last reset or last time collection enabled |
| ExecuteSSCount | Returns number of calls to <i>Service Scripts</i> since last reset or last time collection enabled |
| Operations | |
| CompleteClassPath | Returns the class path used for the Mbeans |
| reset | Resets statistical values. See Resetting Statistics for more advice on this operation. |
| maxTime | Returns maximum (worst case) time, in ms, for the designated service since the last reset or last time collection enabled. |
| minTime | Returns minimum (best case) time, in ms, for the designated service since the last reset or last time collection enabled. |
| completeExecutionDump | 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 | Returns average time, in ms, for the designated service since the last reset or last time collection enabled. |
| executionInfo | 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 | Returns list of services and service types since the last reset or last time collection enabled. See Service Lists for details of format. |
| ProbeTransactionMbean | Component Probe |
| Operations | |
| checkApplicationServer | Constructs HTTP/HTTPS request and returns RFC 2616 HTTP response code and response time in ms |
| checkDatabaseConnectivity | Executes SQL SELECT to test database connectivity. Returns response time in ms. |
| checkBusinessApplicatioServer | Executes base service (read only) to return application server performance in ms. |
| OEMTargetsBean | Application Management Pack for Oracle Utilities interface |

| Mbean | Usage |
|-----------------------|---|
| Operations | |
| toXML | Generate Target information for Oracle Enterprise Manager. An XML document with information for the Application Management Pack for Oracle Utilities. |

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.

Batch JMX Reference **BATCH**

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/batchConnector`

Where:

`<host>` A Batch Server host name

`<jmx_port>` The JMX Port specified using **BATCH_RMI_PORT** from the [ENVIRON.INI](#) configuration file.

*Note: In a multi-host environment, it is possible to reserve a specific host/port combination using the **WEB_BATCH_CLUSTER_URL** which specifies a specific host and **BATCH_RMI_PORT** combination as the administration machine.*

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):

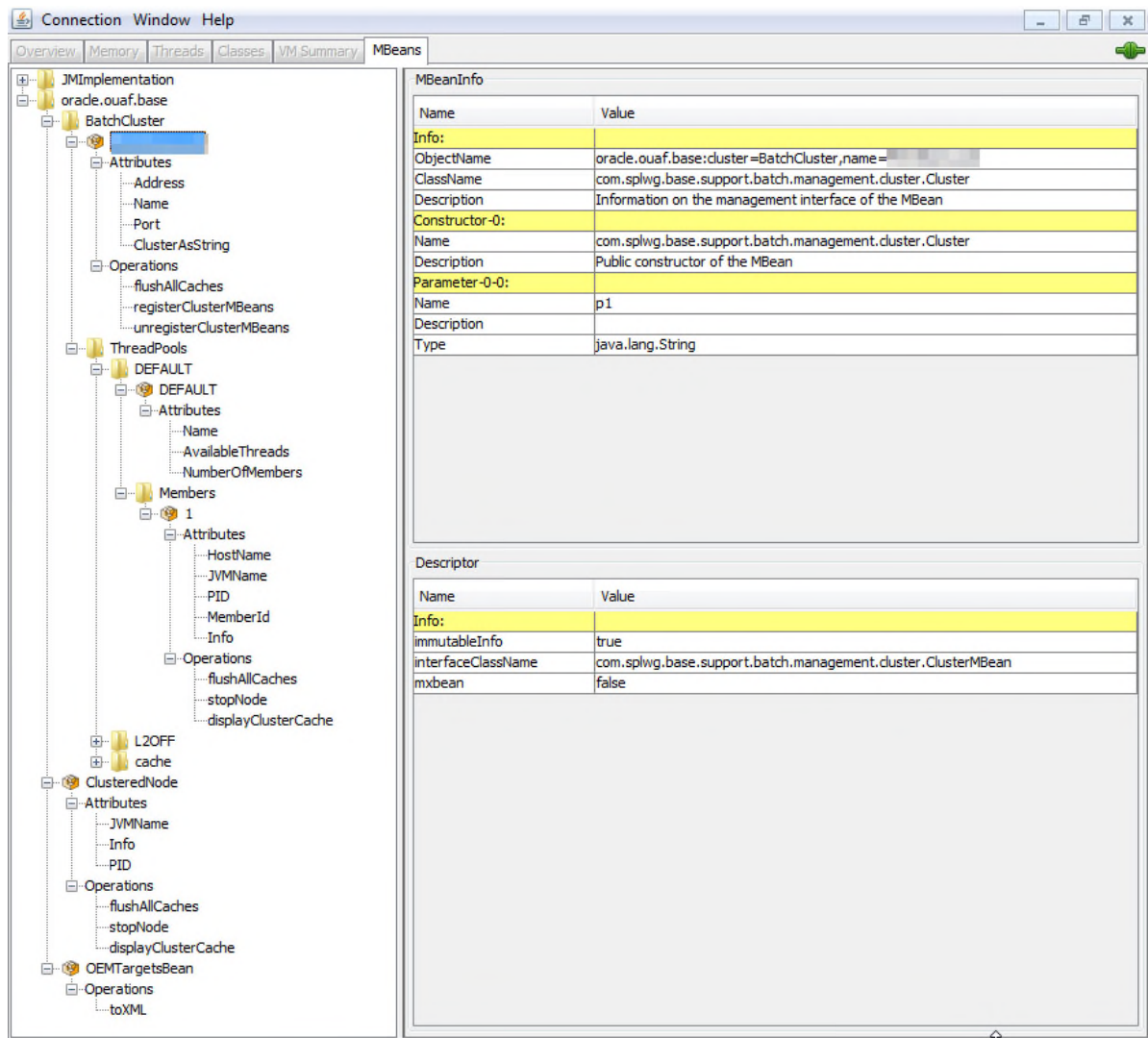


Figure 15 – Example Batch Server Mbeans in jconsole

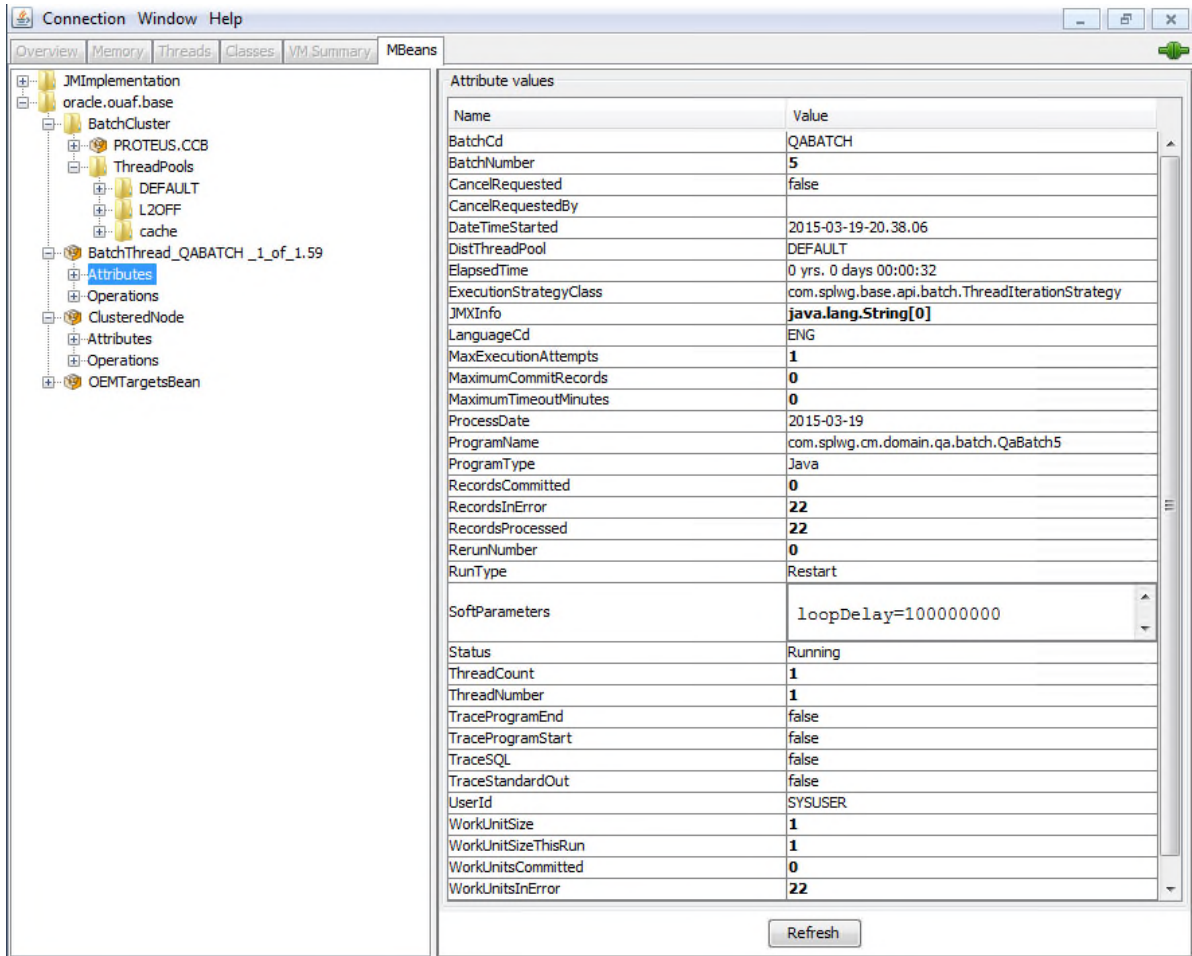


Figure 16 – Example Batch Thread Mbean in jconsole

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

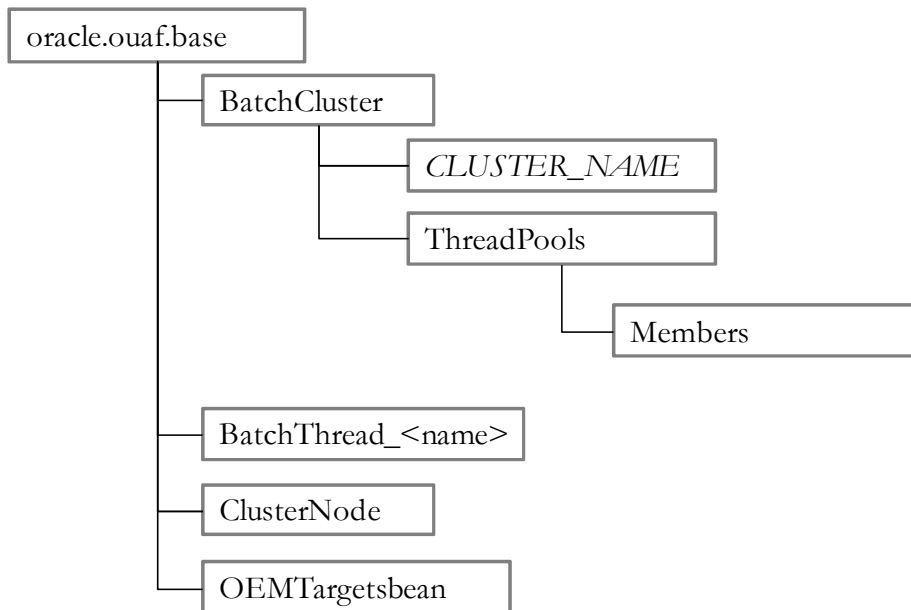


Figure 17 – Batch Server Mbeans

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

| Mbean | Usage |
|--------------------------------|---|
| BatchCluster | Cluster Mbean for cluster with name <ClusterName> |
| Attributes | |
| Address | Cluster address as specified in COHERENCE_CLUSTER_ADDRESS in ENVIRON.INI |
| Name | Name of Cluster as specified in COHERENCE_CLUSTER_NAME in ENVIRON.INI |
| Port | Port number assigned to Cluster as specified in COHERENCE_CLUSTER_PORT in ENVIRON.INI |
| ClusterAsString | XML representation of the cluster for use with Application Management Pack for Oracle Utilities. The structure represents the hierarchy of JMX calls in XML format. |
| Operations | |
| flushAllCaches | Flush the data reuse cache across the batch cluster. Invoke this operation to reload configuration data changes for batch jobs. |
| registerClusterMBeans | Register the Mbeans for lower level tracking. This needs to be invoked to allow threadpool and batch thread level tracking. |
| unregisterClusterMBeans | Disable lower level tracking. This stops low level tracking. |
| ThreadPool¹² | Threadpool Statistics (an entry per threadpool) |
| Attributes | |
| Name | Name of Threadpool |
| AvailableThreads | Number of spare threads for batch processes. A value of zero (0) indicates the threadpool is at capacity. <i>Note: For cache or administration threadpools, this value is always zero (0).</i> |
| NumberOfMembers | Number of members/hosts defined to the threadpool. A value of zero (0) indicates a cache or administration threadpool. |
| Members | Member (submitters/nodes) statistics |
| Attributes | |
| Hostname | Name of Host hosting this threadpool instance |
| JVMName | Name of JVM assigned at runtime |
| PID | Unique OS Process Id for JVM |
| MemberId | Member Number. This number of unique across the cluster and is used to track the member internally by the framework. |
| Info | Parameters used to start threadpool instance in free format. |

¹² This tree will only appear if registerClusterMbeans is executed to enable the additional metrics.

| Mbean | Usage |
|--|--|
| | This is primarily used by Application Management Pack for Oracle Utilities. |
| Operations | |
| flushAllCaches | Flush the cache for this instance |
| stopNode | Stop this member. This allows members to be stopped manually. |
| displayClusterCache | Cluster Cache in free format. This is primarily used by Application Management Pack for Oracle Utilities. |
| ClusterNode | Member information about the member you are connected to |
| Attributes | |
| JVMName | Name of JVM assigned at runtime |
| PID | Unique OS Process Id for JVM |
| Info | Parameters used to start threadpool instance in free format. This is primarily used by Application Management Pack for Oracle Utilities. |
| Operations | |
| flushAllCaches | Flush the cache for this instance |
| stopNode | Stop this member. This allows members to be stopped manually. |
| displayClusterCache | Cluster Cache in free format. This is primarily used by Application Management Pack for Oracle Utilities. |
| BatchThread/BatchJob¹³ | Batch Thread/Job Metrics |
| Attributes | |
| BatchCd | Batch Code |
| BatchNumber | Batch Number |
| cancelRequested | True if the thread has been asked to stop running. See <i>CancelReqeustedBy</i> . |
| cancelRequestedBy | If <i>CancelRequested=true</i> , this will be a string indicating the workstation from where the cancellation was requested. This value will also be logged to the Batch Run Tree. |
| DateTimeStarted | The date and time the batch process was started. |
| DistThreadPool | The thread pool to which this batch process belongs. |
| ElapsedTime | How long the batch process has been running in java format |

¹³ BatchThreads is spawned per thread. BatchJob is spawned for jobs with multiple threads as a control node. BatchJob will not be spawned for single thread jobs.

| Mbean | Usage |
|-------------------------------|--|
| ExecutionStrategyClass | This indicates the commit strategy followed by the program |
| JMXInfo | Additional Tags for batch thread |
| LanguageCD | Language Code used for execution for messages |
| MaxExecutionAttempts | MAX-ERROR value |
| MaximumCommitRecords | Commit Interval for Timeouts (<i>Legacy Support only</i>) |
| MaximumTimeoutMinutes | Timeout Interval (<i>Legacy Support only</i>) |
| ProcessDate | Business Date |
| ProgramName | The program name executed. This is set on the Batch Control . |
| ProgramType | The program type. Value is typically: <i>Java</i> |
| RecordsCommitted | The number of record updates that have been committed to the database. See note below. |
| RecordsInError | The number of records so far in error. This is what will be logged to the Batch Run Tree. See note below. |
| RecordsProcessed | The number of records processed so far. This is what will be logged to the Batch Run Tree. See note below. |
| RerunNumber | Rerun Number if RunType is <i>Rerun</i> . |
| RunType | The type of run: <i>New Run, Restart</i> or <i>Rerun</i> . |
| SoftParameters | List of job parameters from Batch Control . |
| Status | Current status of the thread. Valid values are: <i>Initializing</i> " (at the start of execution to the call to getJobWork in the application class); <i>Getting Work</i> means it is currently in the process of selecting the work units for the batch process; <i>Got Work</i> means it is has successfully selected the work and is in the process of initiating the threads. |
| ThreadCount | Thread Limit |
| ThreadNumber | Thread Number |
| TraceProgramEnd | Whether tracing for Program End is enabled |
| TraceProgramStart | Whether tracing for Program Start is enabled |
| TraceSQL | Whether SQL statement tracing is enabled |
| TraceStandardOut | Whether output tracing is enabled |
| UserId | User used for authorization |
| workUnitSize | The total number of work units for this batch process. For new and restarted runs, this will always contain the total number of work units as selected in the getJobWork method when the batch process was originally started. |

| Mbean | Usage |
|--|---|
| <code>workUnitsSizeThisRun</code> | This is the number of work units for this particular run. For a restarted run, this value will typically be less than the above value; otherwise they will be the same. |
| <code>workUnitsCommitted</code> | The number of work units that have had their work committed. |
| <code>workUnitsInError</code> | The work units that have been found to be in error so far. |
| <code>workUnitsProcessed</code> | The work units that have been processed so far. |
| Operations | |
| <code>cancelThread</code> | Cancel this thread. This updates <code>cancelRequested</code> and <code>cancelRequestedBy</code> . Information is persisted on the Batch Run Tree. |
| OEMTargetsBean | |
| Application Management Pack for Oracle Utilities interface | |
| Operations | |
| <code>toXML</code> | Generate Target information for Oracle Enterprise Manager. An XML document with information for the Application Management Pack for Oracle Utilities. |

Note: Each batch thread is uniquely identifiable in the BatchThread JMX call with the name `<mbean>.<batchcd>.<thread>_of_<threadlimit>.<javathread>`. Where `<mbean>` is BatchJob or BatchThread, `<batchcd>` is the batch control identifier for the job, `<thread>_of_<threadlimit>` is the thread within thread limit and `<javathread>` is the internal java thread number.

Note: The "Records..." numbers are what will be used to log to the Batch Run Tree, and they are usually in step with the "WorkUnits..." values. The reason they are shown separately is because it is possible for custom background processes manually manipulate the record counts for the Batch Run Tree. The true progress status of a thread is reflected in the "WorkUnits..." counts.

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 | <code>scripts/ouaf.jmx.password.file</code> | <code>ouaf.jmx.password.file.template</code> |
| Access Control File | <code>scripts/ouaf.jmx.access.file</code> | <code>ouaf.jmx.access.file.template</code> |

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 allowed 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 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 |

| Column | Comment |
|-------------|---|
| # 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 <i>Clear</i> 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 | Batch |
|---------------------------------------|--|------------------|----------------------------------|
| CLIENT_IDENTIFIER¹⁴ | Userid | Userid | Userid |
| MODULE | Service Name | Web Service Name | Batch Control |
| ACTION | Transaction Type | Transaction Type | Thread Number |
| CLIENT_INFO | Contents Database characteristic User | of Tag on | "Web Service" Threadpool Name |

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.

¹⁴ Due to the length limitation on **CLIENT_IDENTIFIER** the value will be the authorization identifier not the authentication identifier.

Probe Transaction

One of the monitoring facilities in the product is a probe transaction for the online component of the product. This is a set of JMX calls to the various components of the architecture to determine two sets of metrics:

- Availability of the component - The availability will return a pass or fail on availability (including details is possible) of the component.
- Raw performance - The raw transaction performance of the component from the call. This represents the raw performance and should be used as a raw indication of performance at the network level.

The following components are tested using this capability:

| Component | Test | Availability | Performance |
|-----------------------------|-----------------|-------------------|-------------|
| Web Application Server | HTTP/HTTPS ping | RFC 2616 code | Time in ms |
| Business Application Server | Service Call | true or false | Time in ms |
| Database Server | SQL SELECT | True or SQL Error | Time in ms |

The API is part of the [Business Application Server JMX](#) facility.

Note: Customers using Application Management Pack for Oracle Utilities can use additional API's for performance and availability tracking.

Configuration

Configuration Files

There are a number of configuration files that are implemented across the various components of the 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 SPLBASE 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*.

coherence-cache-config.xml - Oracle Coherence Configuration **BATCH**

The [coherence-cache-config.xml](#) configuration file is used by the **CLUSTERED** mode of execution to manage the Oracle Coherence based cache across the batch cluster. This file is generally not altered at the implementation level as it is preconfigured to execute the batch component of the product.

For details of the contents of this file refer to the [Oracle Coherence Integration Guide](#).

e0Batch.properties - General Batch Properties **BATCH**

Note: This configuration file is generated and should not be altered unless instructed by Oracle Support.

This configuration file is used by the batch component to set the location of the installation and the location for any log files generated by the batch component.

This configuration file has two settings that are generated from the [e0Batch.properties.template](#) file located in **templates** directory:

| Configuration Setting | Contents |
|-----------------------|--------------------------------|
| standalone.dir | Location of batch installation |
| SPLOUTPUT | Location of output files |

ejb-jar.xml - Enterprise Java Server Beans Configuration

The Business Application Server and MDB functionality uses Enterprise Java Server beans to implement service calls. By default the behavior of these components is defined and controlled with the [ejb-jar.xml](#) configuration file.

This configuration file is generated by the [initialSetup](#) using the [templates/ejb-jar.xml.template](#) file.

Note: This configuration file is typically NOT altered unless custom JMS or MDB resources are to be included in the servers within the product. Typically JMS resources are not included in the server Refer to the [Oracle WebLogic JMS Integration](#) (Doc Id: [1308181.1](#)) whitepaper for details of JMS or MDB advanced configuration.

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 [Appendix - ENVIRON.INI](#).

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.

hibernate.properties - Database Connectivity Settings

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](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/ucp.html) (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 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 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 .

log4j2.properties – Product 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 *log4j2* Java classes to centralize all log formats into a standard format. The details of the configuration settings and *log4j2* itself are available at <https://logging.apache.org/log4j/2.x/> or <https://logging.apache.org/log4j/2.x/>.

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.

spl.properties – Product Behaviour Settings

The product Web Application has a specific number of settings outside of the JEE 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 JEE 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.

submitbatch.properties - Submitter Configuration **BATCH**

Note: This is the default configuration file. Refer to the [Configuring Batch using bedit](#) section for a discussion of the role of this configuration and its variations

Each background process requires a configuration file to control its behavior. There are two levels of files that control the individual batch job executions:

- [submitbatch.properties](#) - This file sets the global settings for all batch jobs that are submitted. This file is located in `$SPLEBASE/splapp/standalone/config` directory (or `%SPLEBASE%\splapp\standalone\config` on Windows)
- `<batchcode>.properties` - This file is the same format as [submitbatch.properties](#) but contains any job specific settings and/or overrides to settings in the [submitbatch.properties](#).

Note: Any command line options can override settings in any configuration file.

This file is generated using the [bedit](#) facility.

tangosol-coherence-override.xml - Coherence Overrides **BATCH**

The [tangosol-coherence-override.xml](#) configuration file is used by the **CLUSTERED** mode of execution to override the settings for the Oracle Coherence based cache across the batch cluster. This file is generally not altered at the implementation level as it is preconfigured to execute the batch component of the product.

This file is generated using the [bedit](#) facility.

For details of the contents of this file refer to the [Oracle Coherence Integration Guide](#).

threadpoolworker.properties - Threadpool Configuration **BATCH**

Note: This is the default configuration file. Refer to the [Configuring Batch using bedit](#) section for a discussion of the role of this configuration and its variations

Each threadpoolworker requires a configuration file to control its behavior. This file is generated using the [bedit](#) facility for each type of threadpoolworker.

Note: Any command line options on [threadpoolworker\[.sh\]](#) utility can override settings in any configuration file.

This file is generated using the [bedit](#) facility.

web.xml – JEE Deployment Descriptor

Note: For Inbound Web Services, a resource version of this file is generated.

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](#) to build the WAR/EAR file as documented in [Web application server Configuration Process](#)

weblogic.xml – Oracle WebLogic Extensions

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

Note: For Inbound Web Services, a resource version of this file is generated.

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.

| 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 |
| 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 | Defaults from template |

| Parameter | Context | Source |
|------------------------------|---|---|
| | <p>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.</p> <p>A value specified in the console will always take precedence over a manually specified value.</p> | |
| url-rewriting-enabled | <p>Provides methods for configuring a JEE web application that is deployed on an Oracle WebLogic Server instance. Oracle WebLogic Server instantiates this interface only when you deploy a web application.</p> <p>This interface can configure web applications that are deployed as a WAR file or an exploded directory.</p> | 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>
```

weblogic-ejb-jar.xml - WebLogic extensions for Enterprise Java Server Beans

WebLogic Server-specific EJB deployment descriptor that contains elements related to WebLogic Server features such as clustering, caching, and transactions. This file is required if your beans take advantage of WebLogic Server-specific features.

This file is coupled with the [ejb-jar.xml](#) configuration file using the standard [weblogic-ejb-jar.xml](#) schema. This configuration file is generated by the [initialSetup](#) using the [templates/weblogic-ejb-jar.xml.template](#) file.

Note: This configuration file is typically NOT altered unless custom JMS or MDB resources are to be included in the servers within the product. Typically JMS resources are not included in the server Refer to the [Oracle WebLogic JMS Integration](#) (Doc Id: [1308181.1](#)) whitepaper for details of JMS or MDB advanced configuration.

webservices.xml.resource - Web Services configuration

Note: This file is automatically generated and should not be altered manually unless instructed by Oracle Support.

When using Inbound Web Services, a web services deployment descriptor, [webservices.xml](#), must be generated to inform the JEE Web Application Server of the Web Services.

This file is automatically generated from the template file [webservices.xml.resource.iws.template](#) located in the [templates](#) directory.

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. For example:

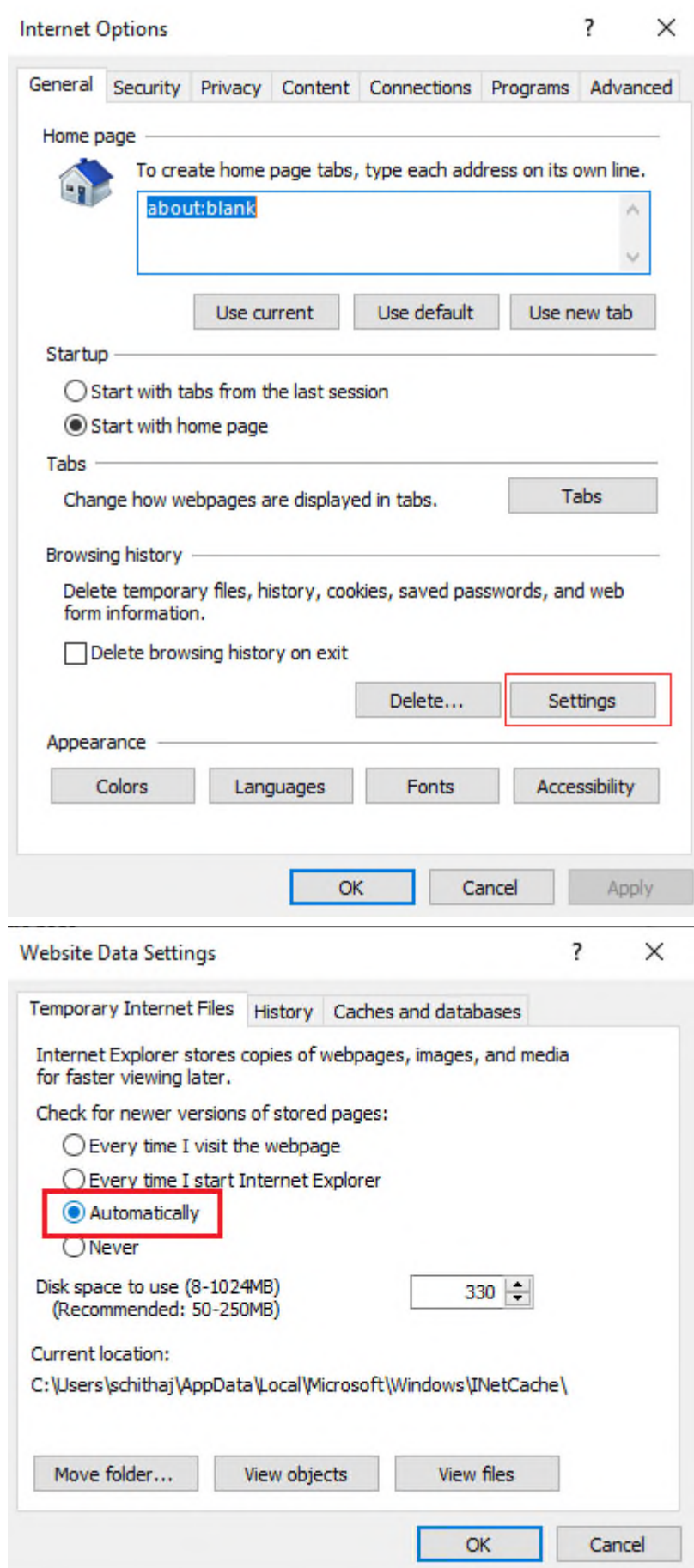


Figure 18 – Internet Explorer Cache Settings

- **Mozilla Firefox/Google Chrome** – 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

- In Microsoft Internet Explorer¹⁵, the product requires support for the HTTP 1.1 protocol to support compression and client cache management.

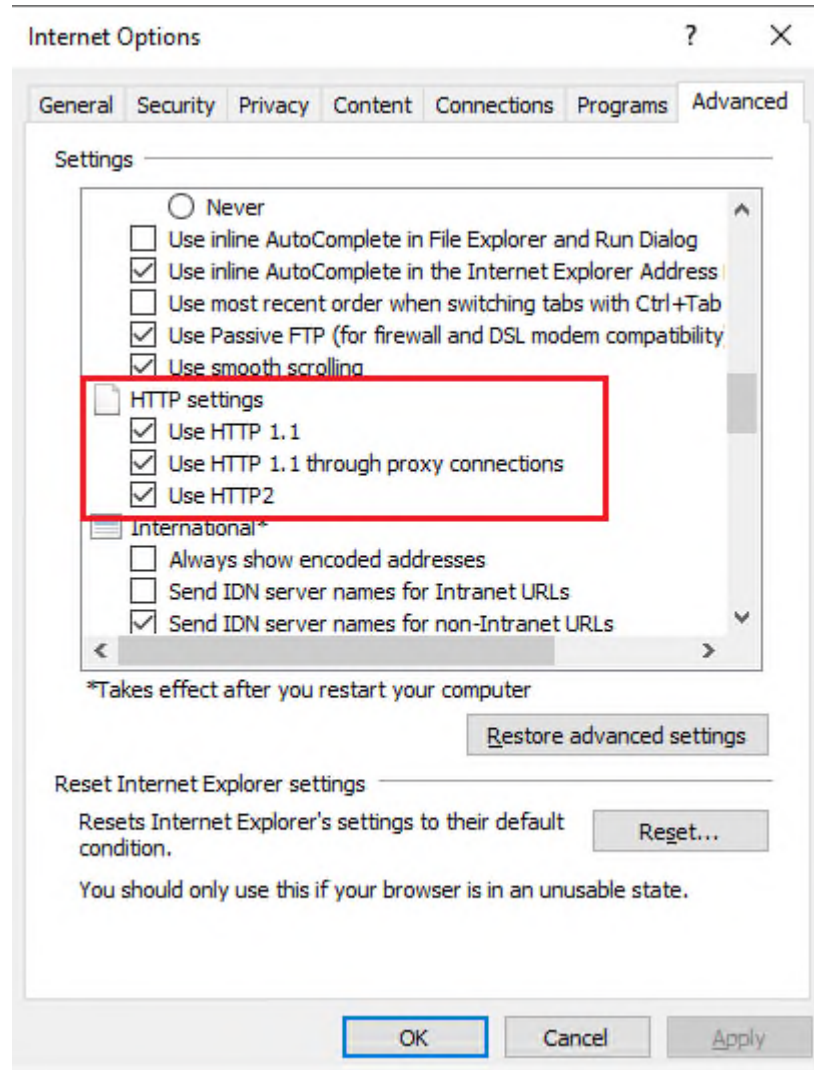


Figure 19 – HTTP 1.1 Settings

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)* in security settings, *Allow sites to run Javascript (Chrome)* in *Content Settings* and *javascript.enabled (Firefox)*¹⁶ in *about:config* must be set to **true**.

¹⁵ This does not apply to Mozilla FireFox and Google Chrome.

¹⁶ By default, Firefox and Chrome enable Javascript automatically.

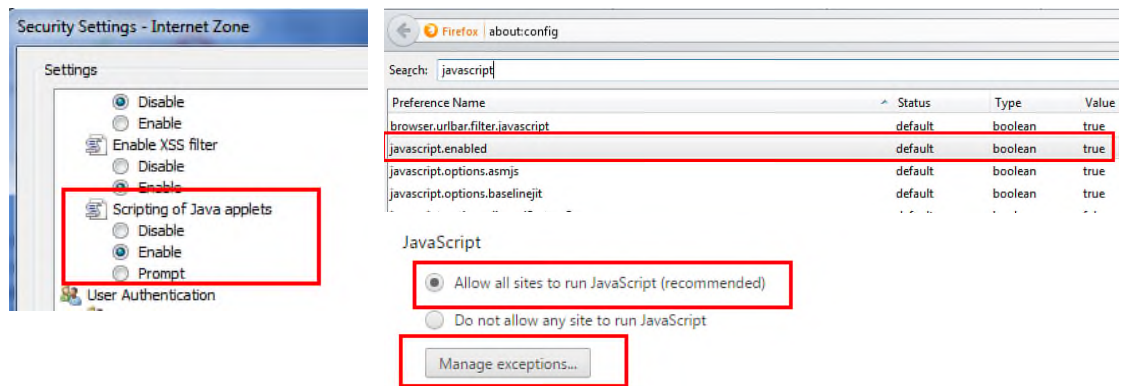


Figure 20 – Javascript Settings

- The product uses popup windows for searches, therefore popup blockers should be configured to allow popups from the product Web application server hosts. Add the Application URL by adding an exception using the relevant function in the browser¹⁷. For example:

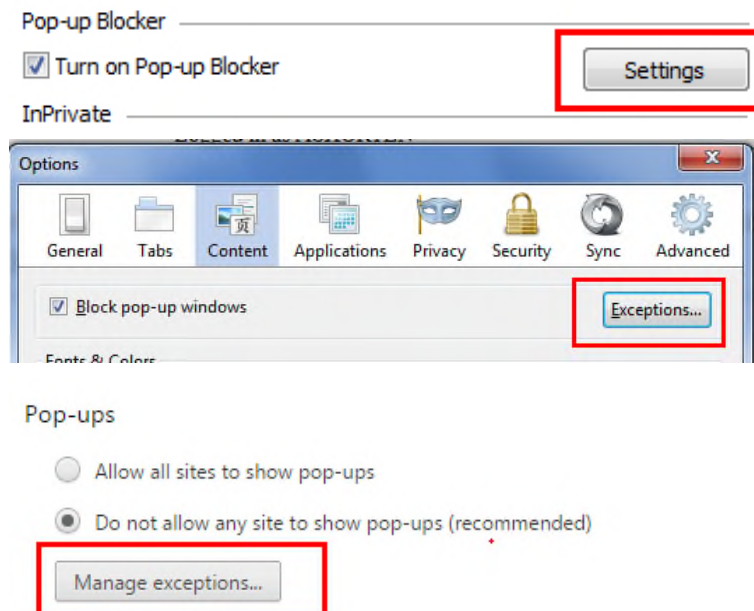


Figure 21 – Popup blocker configuration

- 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 JEE Web application servers. Each JEE 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 JEE Web application server. Refer to the documentation provided with the JEE Web application servers for the location of

¹⁷ Some browsers allow users to add an exception at runtime. Refer to the browser documentation for more information.

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 JEE applications. You will have at least one server per environment. A server uses one Java Virtual Machine (JVM).
- Within a server is the JEE application. It can be a single JEE 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.
- **Webservices** – This is the Web services component.
Note: XAI supported in earlier versions of the product has been replaced by Inbound Web Services. Customers using XAI should migrate to Inbound Web Services. For a discussion of this process refer to [Migrating from XAI to IWS](#) (Doc Id: [1644914.1](#)) available from [My Oracle Support](#).
- **appViewer** – An Application Viewer which contains a data dictionary and source viewer.
- **help** – Online Help.
- **MobileApp** – Mobile Application Platform (**MOB** only)

Web Application Server Configuration Files

Within each JEE Web Application within the JEE 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 JEE specification. In terms of configuration, the product structure within the WAR/EAR file looks like the following:

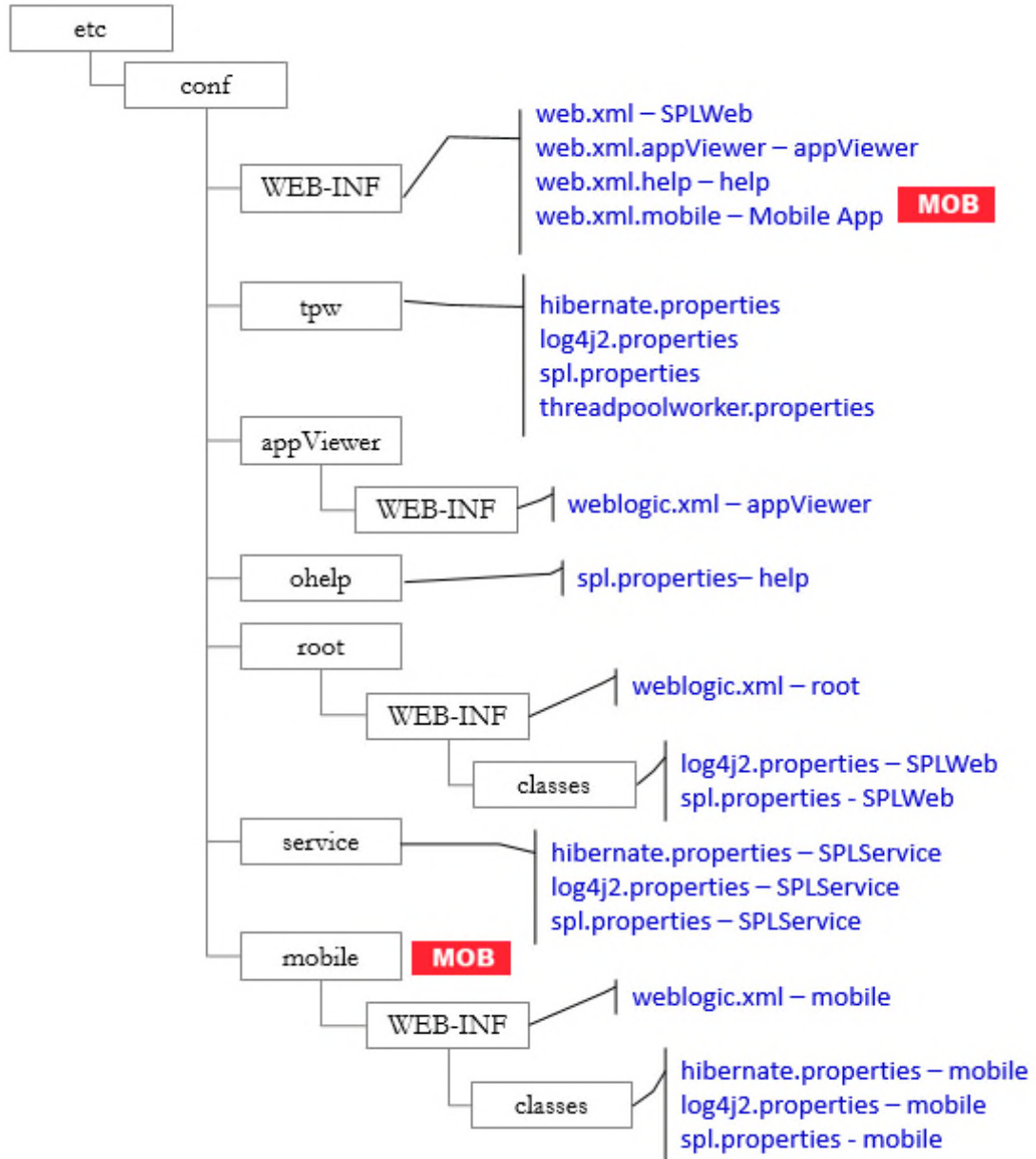


Figure 22 – Configuration File hierarchy

| Configuration File | Contents |
|--------------------------------------|------------------------|
| hibernate.properties | Database connectivity |
| log4j2.properties | Logging configuration |
| spl.properties | Behavior configuration |
| web.xml | Deployment Descriptor |
| weblogic.xml | WebLogic extensions |

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:

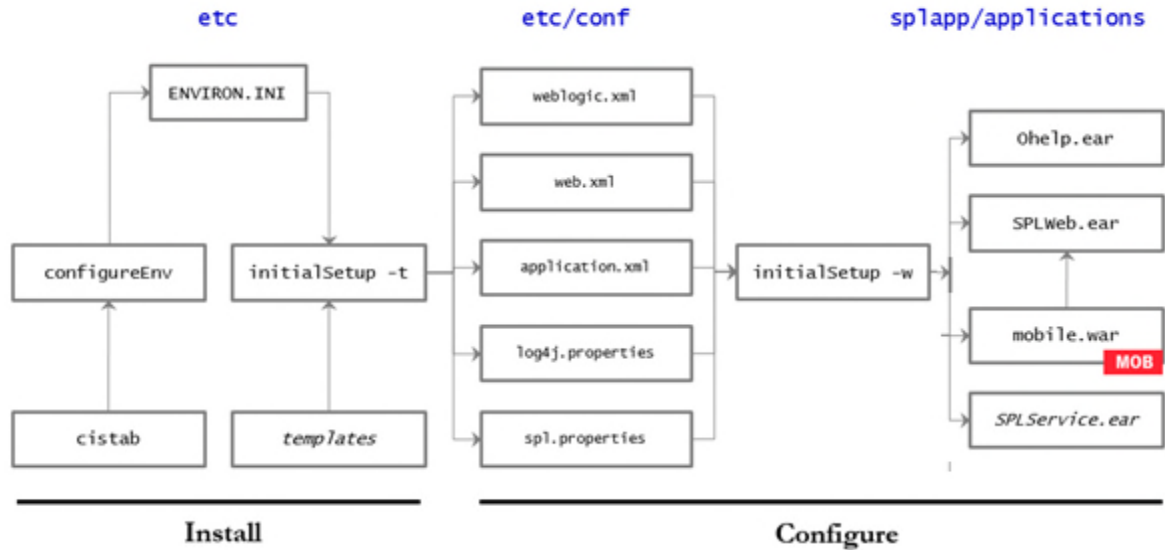


Figure 23 – Web Application Server Configuration Process

- 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](#)
 - [log4j2.properties](#)
 - [spl.properties](#)
 - [weblogic.xml](#)
- 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 | <u>Linux/Unix:</u> |
| Template: | \$SPLEBASE/etc/conf/WEB-INF |
| web.xml.template | <u>Windows:</u> |
| | %SPLEBASE%\etc\conf\WEB-INF |
| spl.properties | <u>Linux/Unix:</u> |

| Configuration File | Destination |
|---|--|
| <u>Template:</u> spl.properties.template | <u>\$SPLEBASE/etc/conf/root/WEB-INF/classes</u> <u>Windows:</u> %SPLEBASE%\etc\conf\root\WEB-INF\classes |
| <u>log4j2.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> log4j2.properties.template | <u>\$SPLEBASE/etc/conf/root/WEB-INF/classes</u> <u>Windows:</u> %SPLEBASE%\etc\conf\root\WEB-INF\classes |
| <u>weblogic.xml</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> weblogic.xml.template | <u>\$SPLEBASE/etc/conf/root/WEB-INF</u> <u>Windows:</u> %SPLEBASE%\etc\conf\root\WEB-INF |
| Application Viewer (appViewer) | |
| <u>web.xml</u> (<i>web.xml.appViewer</i>) | <u>Linux/Unix:</u> |
| <u>Template:</u> web.xml.appviewer.template | <u>\$SPLEBASE/etc/conf/WEB-INF</u> <u>Windows:</u> %SPLEBASE%\etc\conf\WEB-INF |
| <u>weblogic.xml</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> weblogic.xml.appviewer.template | <u>\$SPLEBASE/etc/conf/appViewer/WEB-INF</u> <u>Windows:</u> %SPLEBASE%\etc\conf\appViewer\WEB-INF |
| Help Application (help) | |
| <u>weblogic.xml</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> weblogic.xml.help.template | <u>\$SPLEBASE/etc/conf/help/WEB-INF</u> <u>Windows:</u> %SPLEBASE%\etc\conf\help\WEB-INF |
| <u>spl.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> spl.properties.ohelp.template | <u>\$SPLEBASE/etc/conf/ohelp</u> <u>Windows:</u> %SPLEBASE%\etc\conf\ohelp |
| Mobile Application (MobileApp) MOB | |
| <u>web.xml</u> (<i>web.xml.mobile</i>) | <u>Linux/Unix:</u> |
| <u>Template:</u> web.xml.mobile.template | <u>\$SPLEBASE/etc/conf/WEB-INF</u> <u>Windows:</u> %SPLEBASE%\etc\conf\WEB-INF |
| <u>spl.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> spl.properties.mobile.template | <u>\$SPLEBASE/etc/conf/mobile/WEB-INF/classes</u> <u>Windows:</u> %SPLEBASE%\etc\conf\mobile\WEB-INF\classes |
| <u>log4j2.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> log4j2.properties.mobile.template | <u>\$SPLEBASE/etc/conf/mobile/WEB-INF/classes</u> <u>Windows:</u> |

| Configuration File | Destination |
|--|--|
| | %SPLEBASE%\etc\conf\mobile\WEB-INF\classes |
| weblogic.xml | Linux/Unix: |
| Template: | \$SPLEBASE/etc/conf/mobile/WEB-INF |
| | Windows: |
| weblogic.xml.mobile.template | %SPLEBASE%\etc\conf\mobile\WEB-INF |

- 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 JEE 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 JEE 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 if 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 if you are using configuration overrides.

- If native installation is used, it is recommended to redeploy or update the generated EAR files from the relevant command or console.

User Interface Backward Compatibility Settings

The following settings are used for backward compatibility of the User interface for customers upgrading in the [spl.properties](#) file 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 JEE Web application server.

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

1. Use the deployment utilities provided on the console of the JEE 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 JEE 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 JEE 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 JEE Web application server. Refer to the Installation Guide for specific platform instructions and the administration guide for the JEE Web application server.
3. A number of specific utilities for JEE Web applications are provided with the product to deploy the Web Application to the JEE 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 JEE Web application server. These utilities are outlined below:

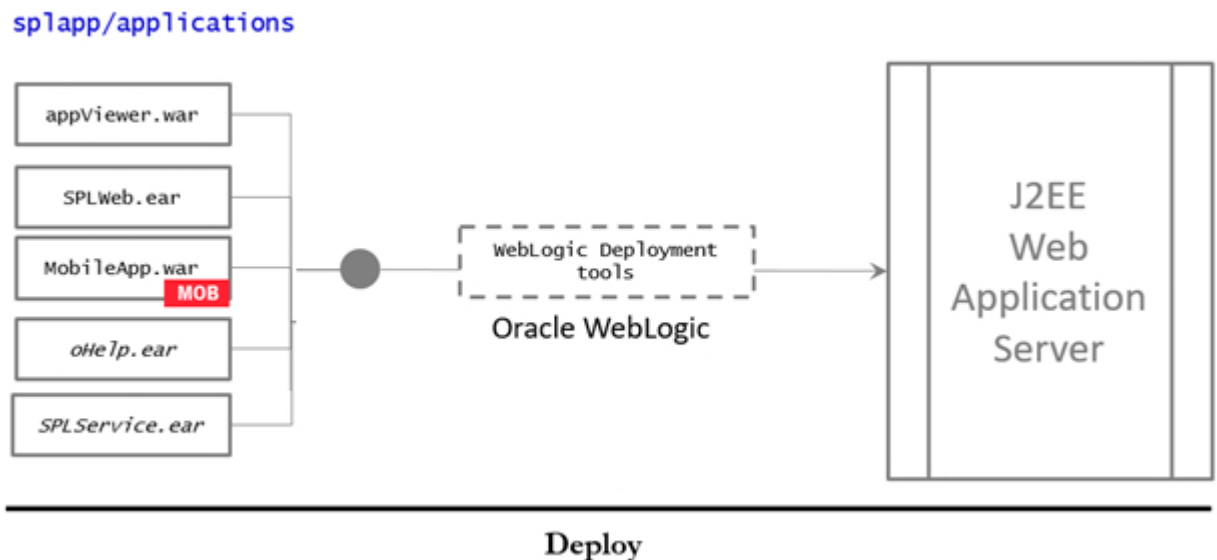


Figure 24 – Web Application Server Deployment Process

- Use the WebLogic console or WLST to deploy/redeploy the EAR files.

This will try to deploy the Web Applications within the JEE Web application server as follows:

| JEE Web application server | Deployment details |
|----------------------------|--|
| Oracle WebLogic | Deployed to WEB_CONTEXT_ROOT application set by SPLWEBAPP to server WEB_WLS_SVRNAME by default using WEB_WLSYSUSER and WEB_WLSYSPASS from the 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** ([SPLWEBAPP](#)) – This contains the Web application server component for

the product.

- **SPLService.ear (SPLSERVICEAPP)** - This contains the Business Application Server component for the product.

There are two modes of installation:

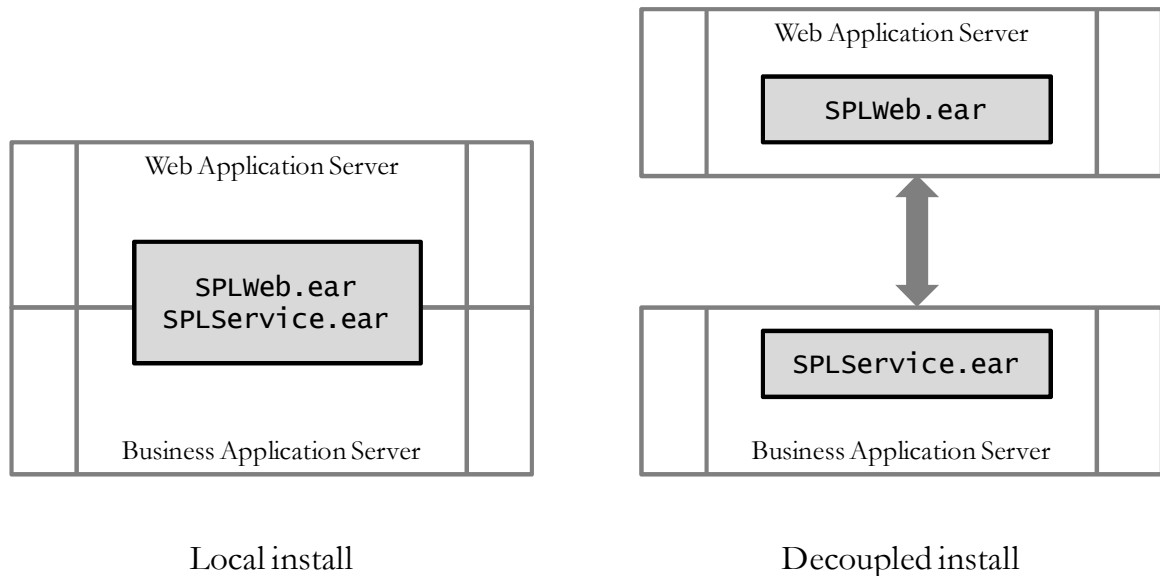


Figure 25 – Installation Modes

- **Local Installation** (also applicable to *expanded* installations for Development environments) - The Web application server and Business Application are on the same instance of the JEE 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 recommended for all environments.

- **Decoupled Installation** – The Business Application Server is on a separate instance of the JEE 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 JEE 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** (or set by **SPLWEBAPP**) EAR file deployed to the Web application server as outlined in [Web Application Server Deployment Process](#) and **SPLService.ear** EAR file (or set by **SPLSERVICEAPP**) 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 JEE 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 Files

Each JEE Web Application within the JEE Web application server has its own configuration files. These files are typically *embedded* within the WAR/EAR files deployed with the product following the JEE 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:

| Configuration File | Contents |
|--------------------------------------|---|
| ejb-jar.xml | Enterprise Service Bean configuration |
| hibernate.properties | Database connectivity |
| log4j2.properties | Logging configuration |
| spl.properties | Behavior configuration |
| web.xml | Deployment Descriptor |
| weblogic.xml | WebLogic extensions |
| weblogic-ejb-jar.xml | WebLogic extensions for Enterprise Service Bean configuration |

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:

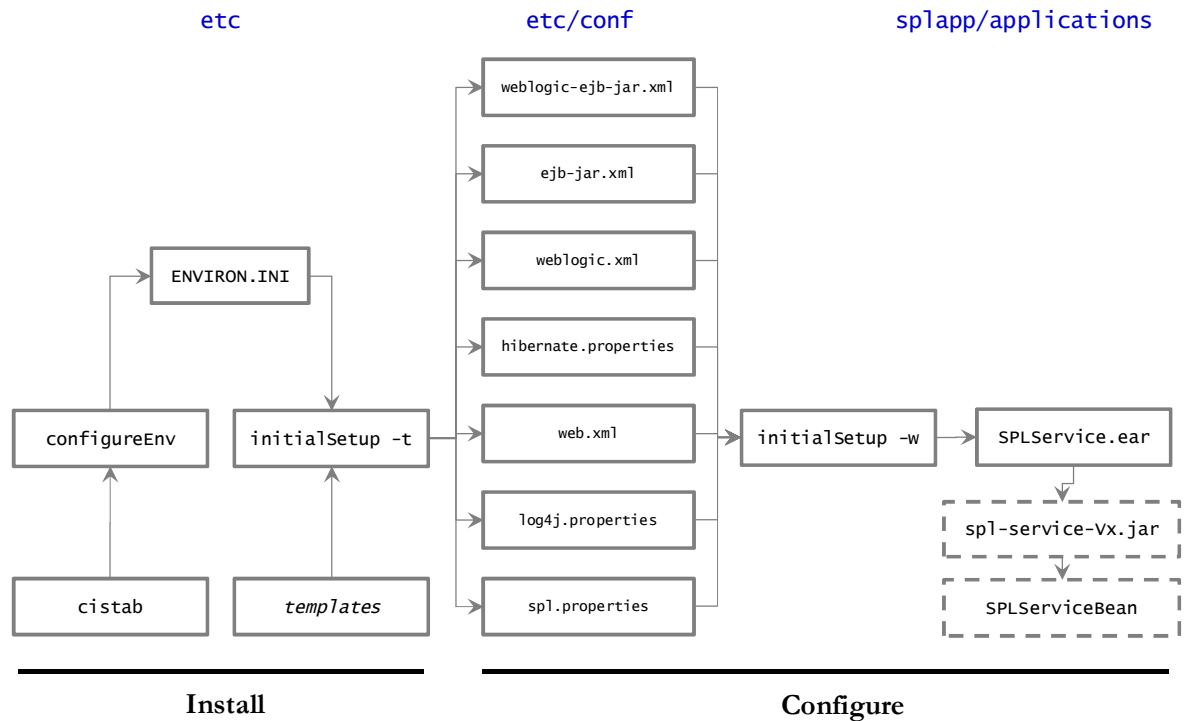


Figure 26 – Business Application Server Configuration Process

- 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](#):
 - [log4j2.properties](#)
 - [spl.properties](#)
 - [hibernate.properties](#)
 - [web.xml](#)
 - [weblogic.xml](#)
 - [ejb-jar.xml](#)
 - [weblogic-ejb-jar.xml](#)
- 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 | Linux/Unix: |
| Template: | \$SPLEBASE/etc/conf/WEB-INF |
| | Windows: |

| Configuration File | Destination |
|--|---|
| web.xml.template | %SPLEBASE%\etc\conf\WEB-INF |
| spl.properties | Linux/Unix: |
| <u>Template:</u> | \$SPLEBASE/etc/conf/service \$SPLEBASE/splapp/businessapp/properties |
| spl.properties.service.template | Windows: |
| | %SPLEBASE%\etc\conf\service %SPLEBASE%\splapp\businessapp\properties |
| log4j2.properties | Linux/Unix: |
| <u>Template:</u> | \$SPLEBASE/etc/conf/service \$SPLEBASE/splapp/businessapp/properties |
| log4j2.properties.service.template | Windows: |
| | %SPLEBASE%\etc\conf\service %SPLEBASE%\splapp\businessapp\properties |
| hibernate.properties | Linux/Unix: |
| <u>Template:</u> | \$SPLEBASE/etc/conf/service |
| hibernate.properties.web.template | Windows: |
| | %SPLEBASE%\etc\conf\service |
| ejb-jar.xml | Linux/Unix: |
| <u>Template:</u> | \$SPLEBASE/splapp/businessapp/config/META-INF |
| ejb-jar.xml.template | Windows: |
| | %SPLEBASE%\splapp\businessapp\config\META-INF |
| weblogic-ejb-jar.xml | Linux/Unix: |
| <u>Template:</u> | \$SPLEBASE/splapp/businessapp/config/META-INF |
| weblogic-ejb-jar.xml.template | Windows: |
| | %SPLEBASE%\splapp\businessapp\config\META-INF |

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

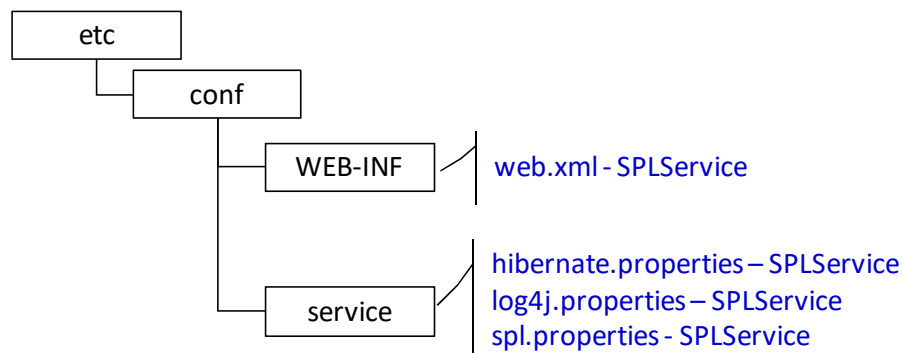


Figure 27 – Business Application Server Configuration files

- 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 (or name set by `SPLSERVICEAPP`) to be deployed into the JEE 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 JEE 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 if 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 if you are using configuration overrides.
- If a native installation

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 JEE Web application server.

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

- Use the deployment utilities provided on the console of the JEE 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 JEE Web application server.

- Use the deployment utilities provided on the command line of the JEE 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 JEE Web application server vendor supplied deployment command line utilities. Refer to the *Installation Guide* for specific platform instructions and the administration guide for the JEE Web application server.
- A number of specific utilities for JEE Web Application are provided with the product to deploy the EJB Application to the JEE 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 JEE Web application server. These utilities are outlined below:

splapp/applications

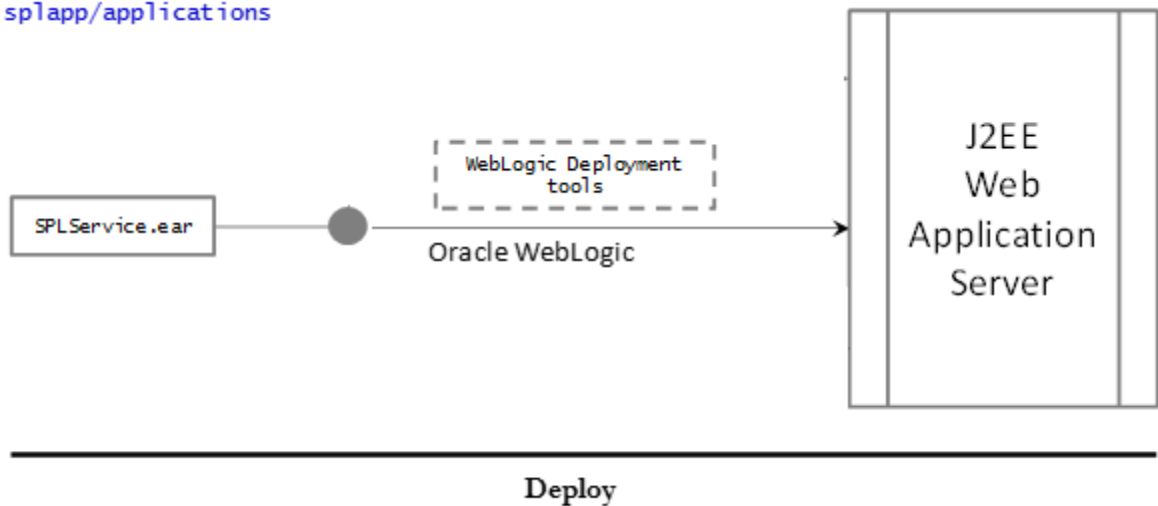


Figure 28 – Business Application Server Deployment process

- Use the WebLogic console or WLST to deploy/redeploy the EAR files.

This will attempt to deploy the EJB Applications within the JEE Web application server as follows:

| JEE Web application server | Deployment details |
|----------------------------|--|
| Oracle WebLogic | Deployed to WEB_CONTEXT_ROOT application set by SPLSERVICEAPP to server WEB_WLS_SVRNAME by default using WEB_WLSYSUSER and WEB_WLSYSPASS from the ENVIRON.INI as administration credentials. |

Inbound Web Services Configuration

The Oracle Utilities Market Settlements Management product supports the ability to implement product Web Services within the JEE 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 JEE Web Application Server directly.
- Individual Inbound Web Services can be monitored from the JEE 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 JEE 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 required to use Inbound Web Services instead of XML Application Integration as the XAI Servlet has been deprecated.

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`¹⁸ from the `ENVIRON.INI`.

Inbound Web Services Configuration Files

The location of the Inbound Web Service is as follows:

¹⁸ `BATCH_DBUSER` is used for command line deployment.

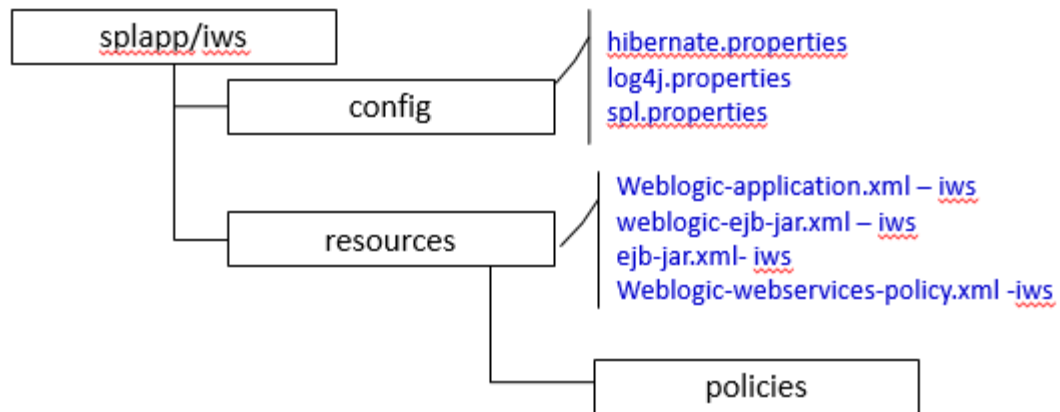


Figure 29 – Inbound Web Services configuration

The following configuration files apply to the Inbound Web Service:

| File | Contents |
|---|--|
| hibernate.properties | Database connectivity for Inbound Web Services |
| log4j2.properties | Logging configuration |
| spl.properties | Inbound Web Service specific configurations |
| weblogic-application.xml | WebLogic extensions. |
| weblogic-ejb-jar.xml | WebLogic extensions for Inbound Web Services |
| weblogic-webservices-policy.xml | Authentication policies applicable for IWS |

Inbound Web Services Concepts

The Inbound Web Services deploys a web archive (**webServices.ear** or name set by [IWSWAR](#)) containing the components of the inbound Web Services using the following structures:

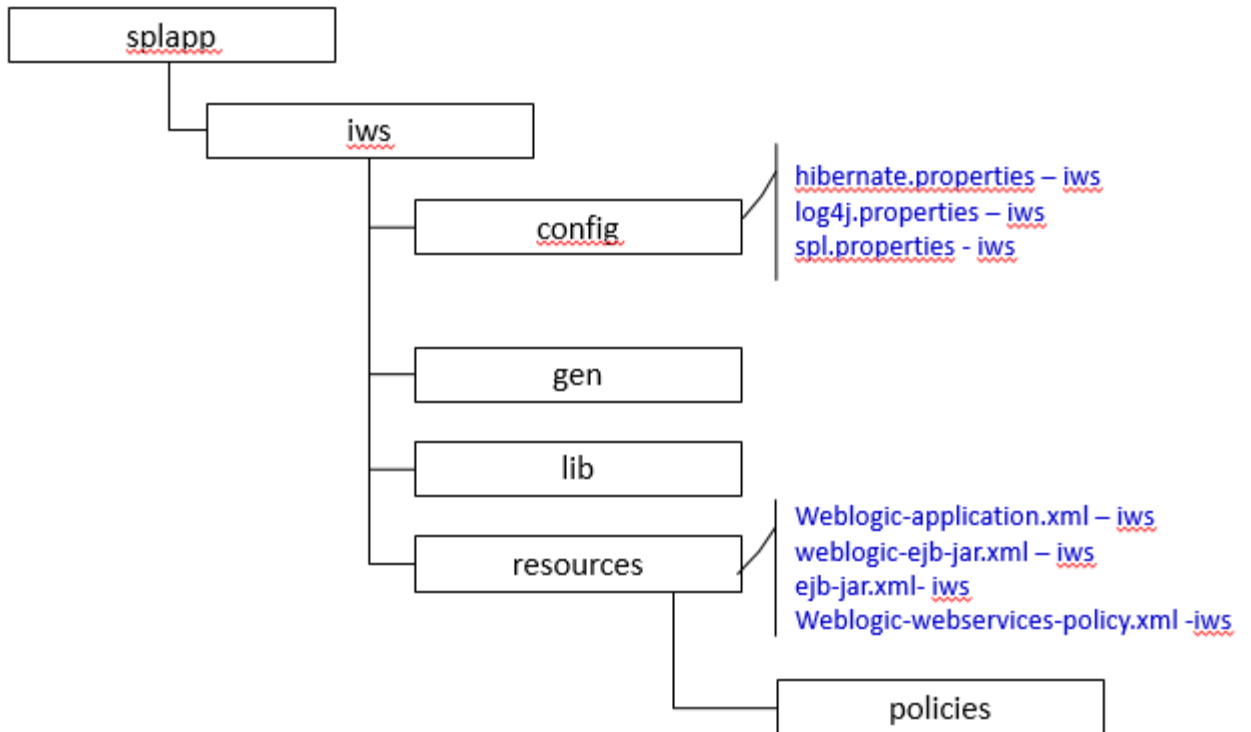


Figure 30 – Inbound Web Services Structures

| Structure | Contents |
|------------------|---|
| config | Configuration files for Inbound Web Services |
| gen | Location of files generated by iwsdeploy |
| lib | Java Libraries |
| resources | Configuration files , custom policy, XSL and XSD files |
| policies | Configuration files defining security policies for 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.ear** file (or name set by [IWSWAR](#)) 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.

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. Refer to the [Migrating XAI to IWS](#) (Doc Id: [1644914.1](#)) available from [My Oracle Support](#) for a discussion of Inbound Web Services.

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:

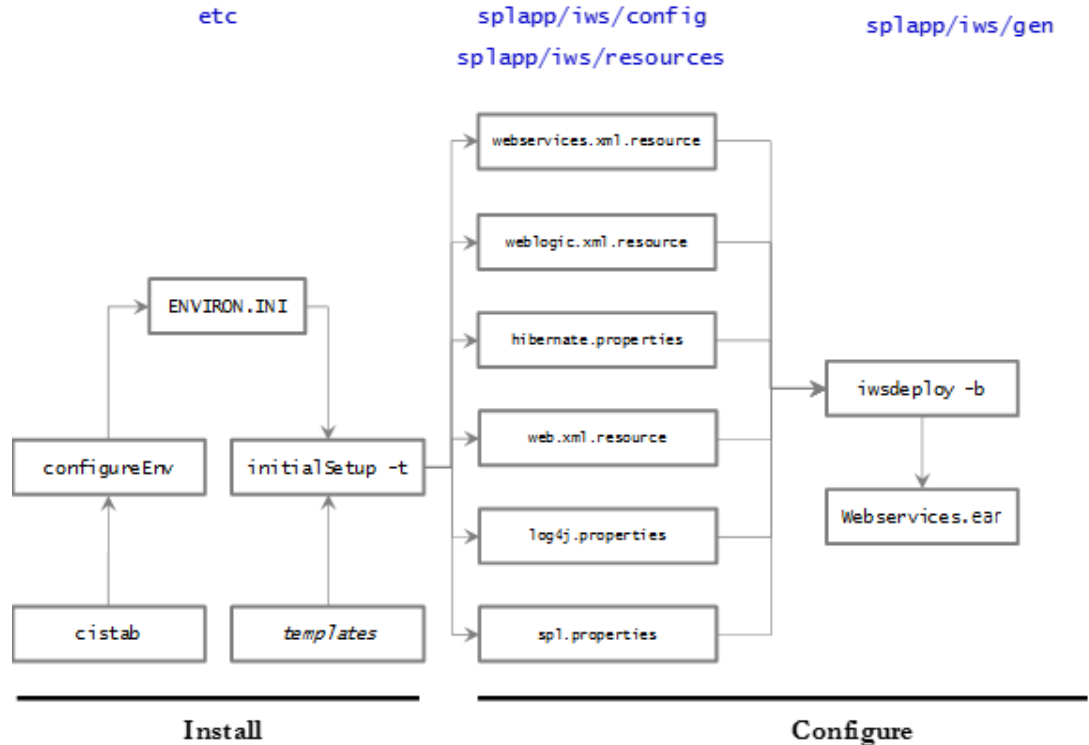


Figure 31 – Inbound Web Services Configuration Process

- 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](#)
- 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 | |
| spl.properties | Linux/Unix: |
| Template: | <code>\$SPLEBASE/splapp/iws/config</code> |
| <code>spl.properties.iws.template</code> | Windows: |
| | <code>%SPLEBASE%\splapp\iws\config</code> |

| Configuration File | Destination |
|---|---------------------------------|
| <u>log4j2.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> | \$SPLEBASE/splapp/iws/config |
| log4j2.properties.iws.template | <u>Windows:</u> |
| | %SPLEBASE%\splapp\iws\config |
| <u>hibernate.properties</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> | \$SPLEBASE/splapp/iws/config |
| hibernate.properties.iws.template | <u>Windows:</u> |
| | %SPLEBASE%\splapp\iws\config |
| <u>web.xml.resource</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> | \$SPLEBASE/splapp/iws/resources |
| web.xml.resource.iws.template | <u>Windows:</u> |
| | %SPLEBASE%\splapp\iws\resources |
| <u>webservices.xml.resource</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> | \$SPLEBASE/splapp/iws/resources |
| webservices.xml.resource.iws.template | <u>Windows:</u> |
| | %SPLEBASE%\splapp\iws\resources |
| <u>weblogic.xml.resource</u> | <u>Linux/Unix:</u> |
| <u>Template:</u> | \$SPLEBASE/splapp/iws/resources |
| weblogic.xml.iws.template | <u>Windows:</u> |
| | %SPLEBASE%\splapp\iws\resources |

- 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 **Webseivces.ear** (or the value specified in **IWSWAR** setting) and to be deployed into the JEE Web application server. This step is optional if configuration overrides are in use.

Note: For customers who want to use manual deployment methods, use the [iwsdeploy\[.sh\] -b](#) option to prevent autodeploy.

At this point the product Web Services are ready for deployment into the JEE Web application

server.

Inbound Web Services Deployment Process

After the build of the Inbound Web Services EAR 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 JEE Web application server.

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

- Use the deployment utilities provided on the console of the JEE Web application server. The EAR files is available under `$SPLEBASE/sp1app/iws/gen` (or `%SPLEBASE%\sp1app\iws\gen` for Windows) can be manually deployed using the console. Refer to the *Installation Guide* for specific platform instructions and the administration guide for the JEE Web application server.
- Use the deployment utilities provided on the command line of the JEE Web application server. The EAR files is available under `$SPLEBASE/sp1app/iws/gen` (or `%SPLEBASE%\sp1app\iws\gen` for Windows) can be manually deployed using the JEE Web application server vendor supplied deployment command line utilities. Refer to the *Installation Guide* for specific platform instructions and the administration guide for the JEE Web application server.
- Use the [iwsdeploy](#) utility for command line access for deployment.
- Use the Inbound Web Services online transaction

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

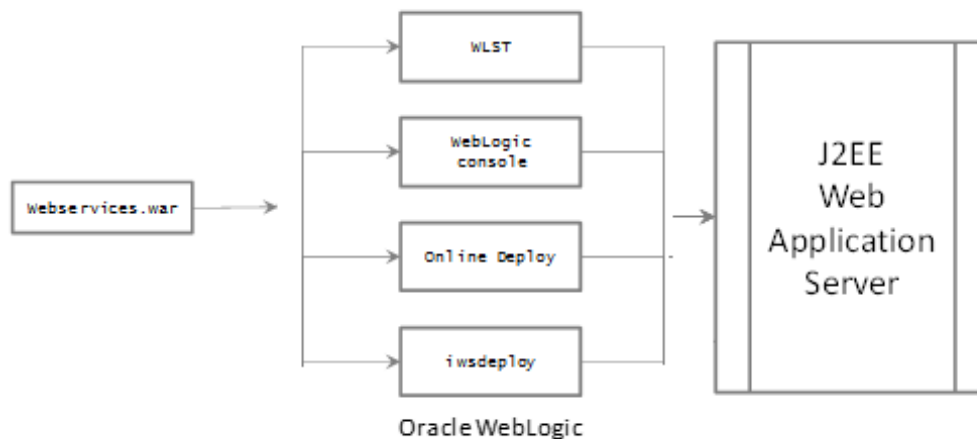


Figure 32 – Inbound Web Services Deployment process

These utilities will attempt to deploy the Web Services within the JEE Web application server as follows:

| JEE Web application server | Deployment details |
|----------------------------|---|
| Oracle WebLogic | Deployed application set by IWSWAR to server WEB_WLS_SVRNAME by default using WEB_WLSYSUSER and WEB_WLSYSPASS from the ENVIRON.INI as administration credentials. |

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 template](#)) 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 if 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.

Inbound Web Services Deployment Utilities

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 JEE Web application server.

There are three methods of deploying the Inbound Web Services component within the JEE 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:

| Web Service Name | Description | Service Revision | Deployment Status | Trace | Debug | Active |
|-------------------------------------|-------------------------------------|------------------|-------------------|-------|-------|--------|
| 1 F1-EmailService | Real-time Email Service | 2 | Deployed | N | N | Y |
| 2 F1-ORAGEOCD | Geocode Address XAI Inbound Service | 3 | Deployed | N | N | Y |
| 3 F1-SmsReceive | Receive SMS Message XAI Service | 3 | Deployed | N | N | Y |
| 4 F1-UpdateAndTransitionSyncRequest | F1-UpdateSyncRequest | 3 | Deployed | N | N | Y |

Figure 33 – Inbound Web Services online transaction

- Use the [iwsdeploy](#) utility to trigger a deployment from the command line.
- Execute the [iwsdeploy](#) utility but manually deploy the Web Services EAR file to the server(s) or cluster using the JEE console or deployment utilities.

Batch Server Configuration **BATCH**

The batch component contains a number of configuration files and utilities to manage the configuration of the cluster, threadpools and submitters. This section outlines the configuration files and configuration process for configuring batch.

Batch Configuration Files

As with the online component of the Oracle Utilities Application Framework there are a number of configuration files that control the performance and behaviour of the batch component.

The batch component houses the configuration files differently to the online and web services component. The online and web services are housed within a JEE Web Application Server and therefore the configuration files are located according to the JEE standards.

In the batch component the configuration are housed in directories as the batch component is housed as a standalone process that is clustered using Oracle Coherence. Therefore, during the configuration process, the configuration files used by the batch component are built using templates using the [initialSetup](#) utility. This utility deposits the configuration files in the `$SPLEBASE\splapp\standalone\config` directory (or `%SPLEBASE%/splapp/standalone/config` directory in Windows).

The figure below summarizes the directory structure and the relevant configuration files:

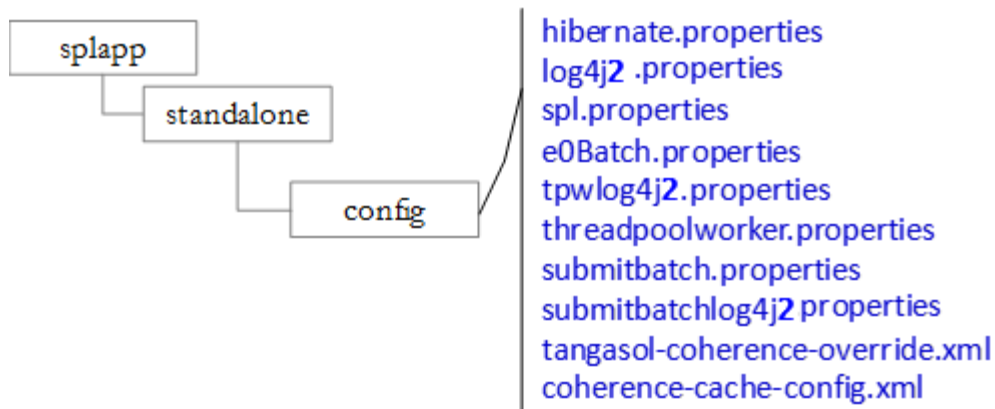


Figure 34 – Batch configuration files

The following configuration files (along with their templates) are listed below:

| Configuration File | Contents |
|---|-----------------------------------|
| e0Batch.properties | General environment settings |
| hibernate.properties | Database connectivity |
| log4j2.properties | Logging settings |
| spl.properties | Application behaviour |
| submitbatch.properties | submitjob default settings |
| submitbatchlog4j2.properties | submitjob logging settings |
| threadpoolworker.properties | threadpoolworker configuration |
| tpwlog4j2.properties | threadpoolworker logging settings |
| coherence-cache-config.xml | Cache settings for cluster |
| tangasol-coherence-override.xml | Override setting for cluster |

General Configuration Process

To configure the batch component during the installation process and post-installation then the following process should be used:

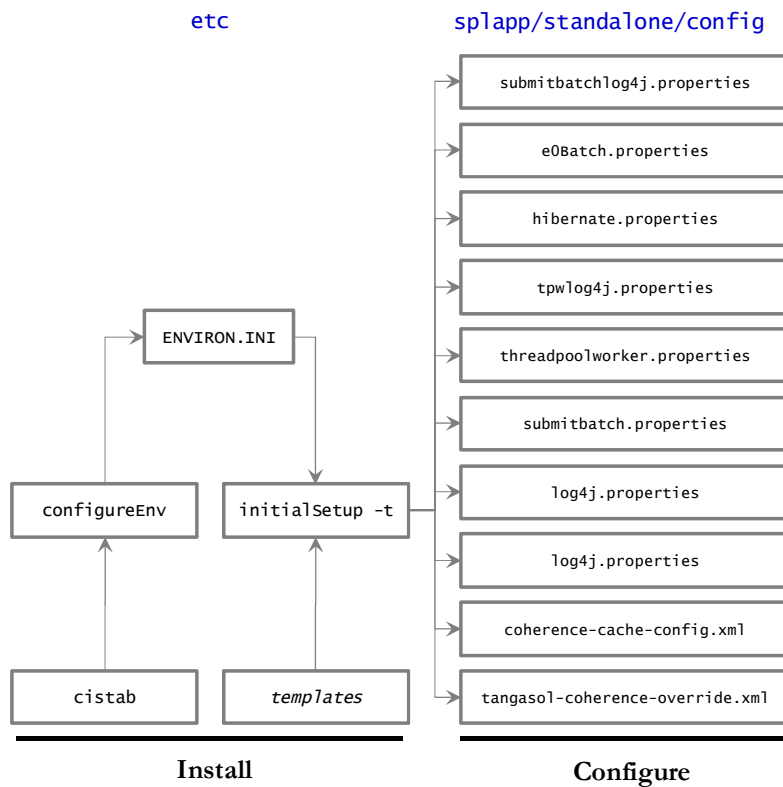


Figure 35 – Batch configuration process

- The [configureEnv](#) utility is used during installation time and can be used post installation to set parameters in the [ENVIRON.INI](#).

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 batch component using the [initialSetup](#) utility. The [initialSetup](#) utility takes the relevant templates, builds the configuration files and deposits them in the `$SPLEBASE\splapp\standalone\config` directory (or `%SPLEBASE%/splapp/standalone/config` directory in Windows).

The configuration files are now ready to be used for the batch component. Additional configuration can be performed using [bedit](#).

Configuring Batch using bedit

The [bedit](#) utility is a set of wizards that allow a batch configuration to be built for an implementation using simple commands. The utility uses a set of templates to build a complete set of configuration files for the cluster, its threadpools and submitters at a global and individual level.

For guidelines and scenarios using this facility, refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) whitepaper available from [My Oracle Support](#).

Batch Configuration Overview using edit

The [bedit](#) utility allows for a set of configuration files to be built on top of the global configuration setting managed using the [Batch Configuration process](#). The following process

should be used:

- **Global Settings** - Specify global batch settings using the [configureEnv](#) utility to set the defaults in the [ENVIRON.INI](#).
- **Apply Settings** - Apply any configuration settings to the global templates and default configuration files using the [initialSetup](#)¹⁹ utility.
- **Perform Cluster Configuration** - Configure the desired [cluster configuration](#) using [bedit](#). Clusters are setup once per environment in either using multicast, unicast or single server optimized configurations.
- **Perform Threadpoolworker Configuration** - Configure the desired [threadpoolworker configuration](#) using [bedit](#). Threadpoolworkers can be created on an individual basis with unique configurations or *labelled* to share configurations.
- **Perform Submitter Configuration** - Configure the desired [submitter configuration](#) using [bedit](#). Submitters can be setup for specific development execution modes (**THIN** or **LOCAL**), generic configuration or project specific jobs.

The process is shown below:

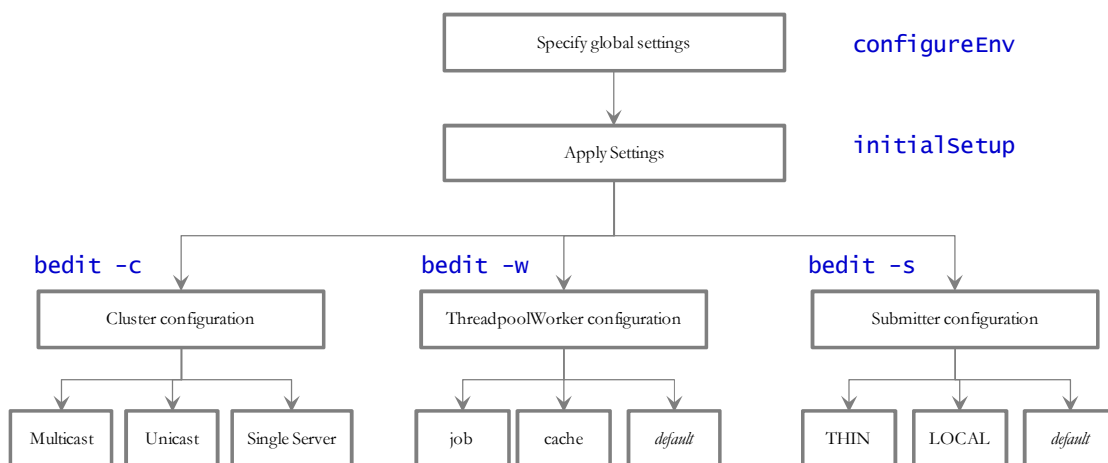


Figure 36 – bedit configuration process

*Note: At any time the file name and location of the generated file is determined at the start of the edit session or using the **what** command.*

Batch Edit Usage Guidelines

The [bedit](#) utility is available to simplify the creation and maintenance of these files to promote stability and flexibility whilst minimizing maintenance mistakes. The following guidelines apply when using the utility:

- The command line options specify the configuration file to be modified. The specific configuration file or the shortcut option can be used to set the scope of the command.
- Pre-optimized specialist templates (with **.be** extension) are used to decide the parameters applicable to the setting or command used in the configuration file being edited. The template name is displayed at the start of execution to confirm the configuration file edited.

¹⁹ Use the **-t** option to allow templates to be rebuilt only. Using other options may affect online.

- Configuration settings can be added, modified or deleted using interactive commands. The following commands are supported:

| Command | Usage |
|--------------------------|--|
| add <group> | Add a group <group> with all its sub-elements. Values for the variables are defaulted from the template. |
| del <group> | Delete the group <group> and all its sub-elements. |
| exit | Exit the editor. If changes have been made, confirmation may be required. |
| help <topic> | Show online help for command or setting. Without a topic a list of topics for the configuration file and commands are listed |
| save | Save any changes to the configuration file |
| set <var> <value> | This command to update a variable's <var> to value <value>. If there are multiple sub-elements for a group, the variable name must be fully qualified by its parent group(s) - e.g. set pool.2 poolname DEFAULT . If there is only one group, the variable does not need to be qualified - e.g. set poolname DEFAULT . |
| what | This command shows which file is being edited and the template on which it is based. |

- The templates will generate the necessary settings based upon the scope of the command. Use the online help to determine valid settings for your configuration.
- User preferences for labels and cluster types are stored in the **be.prefs** file.
- Using the **save** command will reflect your changes to the configuration file directly.
- If exit command is issued and changes have been detected then a confirmation message will be issued. Press **y** to save, **n** to quit without saving or **c** to cancel out of exit command.

Each subsection will outline specific settings within the scope of each configuration.

Cluster Configuration using **bedit**

Oracle Coherence is used to define and manage the batch cluster. The cluster must be configured to decide how the cluster will operate on the network and the scope of the cluster. The major decision for this step in the process is the type of cluster to implement. There are three styles of cluster supported:

- **Multicast cluster** (default) - A batch cluster based upon the multicast network protocol. The protocol requires a multicast address and common port. This protocol supports dynamic clustering.
- **Unicast cluster** - A batch cluster based upon unicast network protocol. Each host and port in the cluster must be explicitly configured and each server must be.
- **Single Server cluster** - A batch cluster confined to a single server using unicast. This is designed for use in non-production environments to simplify the configuration.

The templates for these cluster types are as follows:

| Template | Usage |
|---|--|
| <code>tangosol-coherence-override.mc.be</code> | Multicast cluster |
| <code>tangosol-coherence-override.ss.be</code> | Single Server cluster (useful for non-production environments) |
| <code>tangosol-coherence-override.wka.be</code> | Unicast (or wka) cluster |

To configure the cluster use one of the example commands:

| | |
|---|--|
| <code>bedit.sh -c -t <type></code> | Create or edit Cluster configuration. The <code>-t</code> option is used to set the cluster type as specified (<code>mc</code> - Multicast, <code>wka</code> - Unicast or <code>ss</code> - Single Server). Specifying the <code>-t</code> option on an already defined cluster may change the type |
| <code>bedit.sh -c</code> | Maintain the cluster configuration. Cluster type is defaulted from <code>be.prefs</code> . |
| <code>bedit.sh tangosol-coherence-override.xml</code> | Edit the <code>tangosol-coherence-override.xml</code> file. |

Once the configuration is opened the following settings need to be configured:

| Variable | ss | wka | mc | Recommendations |
|-----------------------|----|-----|----|---|
| <code>cluster</code> | ■ | ■ | ■ | Unique cluster name for environment. This must be unique for the cluster. Typically use the environment name or database name. |
| <code>address</code> | ■ | ■ | ■ | IP address or host name for this node in the cluster. Use localhost if possible to minimize maintenance across hosts. |
| <code>port</code> | ■ | ■ | ■ | Unique port used for cluster. This must be unique per cluster per host. It use will vary from cluster type to cluster type. Refer to the online help for more information. |
| <code>loglevel</code> | ■ | ■ | ■ | The logging level associated with cluster operations. Refer to the online help to decide the amount of information you wish to log. The higher the value the more that is logged. High values are used by developers typically. |
| <code>mode</code> | ■ | ■ | ■ | The Coherence mode that the cluster is to be scoped to. Refer to the online help for more information. |
| <code>socket</code> | | ■ | | This is a section for each of the hosts in the Well Known Address format. Each host is a separate socket entry. Refer to the online help |

| Variable | ss | wka | mc | Recommendations |
|-------------------|----|-----|----|---|
| | | | | for more information. |
| wkaaddress | | ■ | | The IP address or host name of the member of the cluster assigned to this socket. |
| wkport | | ■ | | The port number assigned of the member of the cluster assigned to this socket. This value ideally is the same across all hosts in the cluster but can be overridden to overcome port conflicts. The port number on each node must match the number assigned to the port value. |

For unicast implementations a socket section must be created per host/port combination defined in the file using the following commands:

- Use the **add socket** command to add a new socket group. The group name generated will be in the form **socket.<number>** where **<number>** is the sequence number. For example:

```
> add socket
...
socket.2
    wkaaddress (localhost)
    wkport (42020)
```
- Set the **wkaaddress** and **wkport** using the **set socket.<number> <variable> <value>** where where **<number>** is the sequence number, **<variable>** is **wkaaddress** or **wkport** and **<value>** is the desired value. For example:

```
> set socket.2 wkport 6402
...
socket.2
    wkaaddress (localhost)
    wkport (6402)
```
- Repeat for each node in the cluster.

Threadpoolworker Configuration using bedit

Once a cluster is defined the threadpools that will execute across that cluster must be defined with their name and attributes. The product supports different types of threadpools that can execute batch processes with the following:

- Cache Threadpools** - Threadpoolworkers that do not execute jobs but act as co-ordination node for network traffic. It is recommended that there be at least one cache threadpool per physical host per cluster. For a discussion of cache threadpools, refer to the [Batch Best Practices](#) (Doc Id: [836362.1](#)) whitepaper available from [My Oracle Support](#). This threadpool type has [tangosol.coherence.distributed.localstorage](#) set to **true**.
- Job Threadpools** - Threadpoolworkers that are optimized to run batch threads with support for additional java options and other cluster specific optimization settings.

- **Default Threadpools** - For backward compatibility, the default template used is pre-optimized with settings from past releases. It is recommended to migrate from this template to job template.

The templates for these threadpool types are as follows:

| Template | Usage |
|---|---------------------------|
| <code>threadpoolworker.be.template</code> | Default template |
| <code>threadpoolworker.cache.be.template</code> | Cache Threadpool template |
| <code>threadpoolworker.job.be.template</code> | Job Threadpool templates |

To configure the cluster use one of the example commands:

| | |
|---|---|
| <code>bedit.sh -w -l <type></code> | Create or edit threadpool configuration. The <code>-l</code> option is used to set the label type as specified (<code>cache</code> - cache or <code>job</code> - Job). |
| <code>bedit.sh -w</code> | Maintain the default threadpool configuration. |
| <code>bedit.sh threadpoolworker.properties</code> | Edit the <code>threadpoolworker.properties</code> file. |

Once the configuration is opened the following settings need to be configured:

| Parameter | Recommendations |
|---------------------------|--|
| <code>minheap</code> | Minimum JVM Heap size |
| <code>maxheap</code> | Maximum JVM Heap size |
| <code>maxperm</code> | Maximum JVM permgen size |
| <code>daemon</code> | Whether the threadpool should run the online submission daemon. This value should be set to false for production environments. |
| <code>dkidisabled</code> | Key insert behavior |
| <code>storage</code> | Whether local storage is enabled. Used for cache threadpools. |
| <code>distthds</code> | Number of internal threads used for cache threadpools. |
| <code>invocthds</code> | Number of internal threads used for invocation services. |
| <code>role</code> | Information role for the threadpools used for monitoring |
| <code>jmxstartport</code> | JMX Start Port |
| <code>pool</code> | Group section for each threadpool |
| <code>poolname</code> | Name of threadpool |
| <code>threads</code> | Maximum number of threads for the threadpool per instance |

For multiple threadpools per configuration use the following process:

- Use the `add pool` command to add a new pool group. The group name generated will be in the form `pool.<number>` where `<number>` is the sequence number. For example:
`> add pool`

```
...
pool.2
  wkaaddress (localhost)
  wkaport (42020)
```

- Set the **poolname** and **threads** using the **set pool.<number> <variable> <value>** where where **<number>** is the sequence number, **<variable>** is **poolname** or **threads** and **<value>** is the desired value.

Submitter Configuration using bedit

The last step in configuration is to configure the submitters. A submitter is a JVM that submits jobs or threads of jobs to a threadpoolworker and waits for the thread to complete. It is responsible for interfacing to the threadpoolworker and also provides an interface to the process that submits the jobs in the first place (e.g. a third party batch scheduler or Oracle Scheduler).

A submitter needs one or more of the following pieces of information:

- It needs a configuration file that defines the parameters to be used for the individual batch process being executed. These can be global configuration files or individual configuration files optimized for a particular batch process.
- Command line options to set or override particular configuration parameters that define the execution parameters for the individual process or thread.

The single execution of a submitter can run a single threaded batch job, a single individual thread of a multi-threaded batch job or all threads of a multi-threaded job.

To create a specific configuration file for a batch job any of the following commands can be used:

- bedit.sh -l <type>** Create or edit submitter configuration. The **-l** option is used to set the label type for development purposes as specified (**THIN** - THIN mode or **LOCAL** - local mode).
- bedit.sh -b <batch_cd>** Create or edit submitter configuration for job **<batch_cd>**.
- bedit.sh -s** Edit default submitter configuration

Once the configuration is opened the following settings need to be configured:

| Parameter | Recommendations |
|-----------------|---|
| poolname | Name of threadpool to execute this submitter within |
| threads | Thread limit of submitter. The number of threads must be equal to or less than the number of threads allocated to executing instances of the threadpool |
| commit | Default commit interval. This overrides the commit interval defined internally for the batch job. |
| user | The userid, defined to the User record, to be used to determine execution permissions and is used for records updated and created by this process. This MUST be a valid defined used. |
| lang | The language pack used for the batch process (default is ENG) |

| Parameter | Recommendations |
|----------------|--|
| storage | This sets whether this node is a storage node. Used for submitters that use THIN mode (for developers). This value is recommended to be set to false for all other submitter types. |
| role | The role used for this submitter. This is used for the JMX monitoring interface as a filter. By default the batch code used for this value. |
| soft | Group section for soft parameters. One section per parameter |
| parm | Parameter name as outlined on Batch Control. |
| value | Value assigned to the parameter |

For multiple parameters per configuration use the following process:

- Use the **add soft** command to add a new parameter group. The group name generated will be in the form **soft.<number>** where **<number>** is the sequence number. For example:

```
> add soft
...
  soft.2
    parm (maxErrors)
    value (500)
```
- Set the **parm** and **value** using the **set pool.<number> <variable> <value>** where where **<number>** is the sequence number, **<variable>** is **parm** or **value** and **<value>** is the desired value. For example:

```
> set soft.2 parm fred
...
  soft.2
    parm (fred)
    value (500)
```

Batch Configuration Guidelines

The following sections outline general guidelines for batch configuration.

Cache Threadpools

One of the new facilities in the batch architecture is the ability to define cache or storage nodes in your architecture. By default a batch cluster communicates across the cluster elements such as threadpools the state of other elements in the cluster. While this has benefits, as any element in the cluster can be queried, via JMX, to find out the global status of all active processes in that cluster, it can generate a large amounts of network traffic and cause instability as the cluster is performing a large amount of maintenance operations. To address this it is now possible in the product to create cache or storage threadpools. These act as conduits with the architecture and greatly reduce the network traffic and overheads in managing the cluster. This translates to more stable clusters.

The communications between elements are shown below:

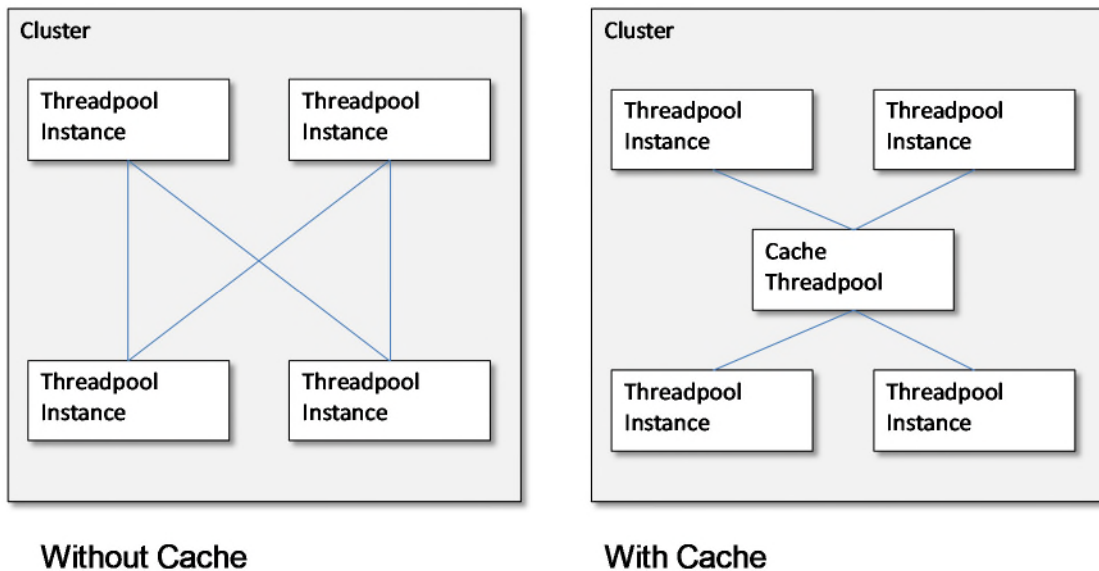


Figure 37 – Cache Architecture

The performance advantages of the cache increases with the number of elements the cluster has to manage and cache threadpools have the following implementation recommendations:

- Cache threadpools do not execute any threads of any jobs within them. They are exclusively used for administration, a storage node for the cluster state and a conduit for cluster management.
- Cache threadpools act as Coherence local storage nodes to maintain the integrity of the cluster and allow cluster management.
- Cache threadpools are ideally suited to allow JMX connections to monitor the performance of the cluster using the [Batch JMX interface](#).
- At least one cache threadpool per cluster per host is recommended. Multiple cache threadpools can be implemented where high availability is required or there are a large number of submitters, threads and/or threadpools to manage.
- If a cache threadpool is shut down and no cache threadpools are active at any time, the cluster will not revert to individual elements communicating across other elements.
- To create cache clusters, use the `bedit[.sh] -l cache` command. A prebuilt template is created for the cache where `storage` is enabled, `distthds`, `invocthds` and the number of threads is set to 0 (to prevent jobs from actually running in the cache).

Adding custom JMX Information to jobs

The [Batch Server JMX interface](#) can be extended for additional tags using a new parameter `f1.jmxInfo` for filtering purposes or additional documentation. For example:

| | |
|------------------------|--|
| ExecutionStrategyClass | com.splwg.base.api.batch.SingleTransactionStrategy |
| JMXInfo | foo2=bar2 foo1=bar1 |
| LanguageCd | ENG |
| MaxExecutionAttempts | 1 |

Figure 38 – Example JMXInfo in jconsole

The `submitbatch.properties` file for the particular job should have the following

additional parameter:

```
com.splwg.batch.submitter.softParameter.f1.jmxInfo.<tag>=<value>
```

where:

| | |
|----------------------------|---------------------------------------|
| <code><tag></code> | Additional JMX information identifier |
| <code><value></code> | Value for parameter |

Multiple parameters of this type are supported within a single configuration file.

For example:

```
com.splwg.batch.submitter.softParameter.f1.jmxInfo.foo=bar
```

When using `edit` the command sequence to use this facility is:

- Add a soft parameter section for each desired tag/value pair using the command (a section `soft.<sequence>` is created where `<sequence>` is a generated sequence number):
`add soft`
- Specify the parm property as `f1.jmxInfo.<tag>` where `<tag>` is the desired tag using the command:
`set soft.<sequence> parm f1.jmxInfo.<tag>`
- Specify the value property as the desired value for the tag using the command:
`set soft.<sequence> value <value>`

It is possible to specify this setting on the `submitjob[.sh]` command using the `-x` option.

For example:

```
submit.sh ... -x f1.jmxInfo.foo=bar
```

Oracle Cloud Support

The Oracle Utilities products are progressively being implemented on Oracle Cloud in both Software as a Service (SaaS) and Platform as a Service (PaaS). This has resulted in new cloud based configurations and features. These features are also available for customers with on-premise and hybrid implementations. This section will outline the additional configuration facilities that can be used for on-premise implementations.

Support for host whitelisting

The hosts and port numbers used by the feature sets in the product are typically restricted in scope for on-premise products but on the cloud, the ports and hosts can be restricted to ensure compliance. The product supports the configuration of the hosts and ports, that are accessible from the online system²⁰, can be specified in a whitelist which is validated at configuration time and runtime. The product now supports whitelisting of URIs with schemes such as `t3`, `t3s`, `ldap`, `ldaps`, `jdbc`, etc. URI whitelisting can be enabled by setting the `com.oracle.ouaf.uriValidation.enable` parameter in the `spl.properties` file to `true`. Once enabled, it is possible to limit the values of URL's within the product for key objects with the configuration. This is implemented as a whitelist that can filter on scheme (aka protocol), hosts and ports. These are checked at runtime and can generate an error if they do not adhere to the whitelist.

²⁰ URL's within configuration files are subject to deployment restrictions within the deployment facility itself.

The feature allows for the following:

- Individual scheme, hosts and port combinations can be configured to limit runtime access for specific features.
- Specification of the '*' wildcard is supported for scheme, hosts and ports.

Note: Custom whitelists are not supported on Oracle Utilities SaaS Cloud Services.

There are a number of configuration settings (ENVIRON.INI) that control the *whitelist*:

- **[CLOUD_RESTRICTION_URLS_ENABLE](#)** - Whether the whitelisting feature is used or not (default is **false**).
- **[CLOUD_WHITE_LIST_PATH](#)** - The full path to the whitelist file for Oracle Cloud. This variable should not be populated for on-premise implementations. This is only used if **[CLOUD_RESTRICTION_URLS_ENABLE](#)** is set to **true**.
- **[CLOUD_CUSTOM_WHITE_LIST_PATH](#)** - The full path to a custom whitelist for Oracle Cloud or on-premise implementations. This is only used if **[CLOUD_RESTRICTION_URLS_ENABLE](#)** **CLOUD RESTRICTIONS URLS ENABLE - _En** is set to **true**.

The format of the whitelist is an XML format document:

| Attribute | Comments |
|-----------------------|---|
| <uri> | URI tag |
| <path> | Specific path on the host. |
| <host> | Host name or IP Address for host. Wildcards are supported. IPV4 and IPV6 formats are supported. |
| <port> | Port Number. Wildcards are supported. |
| <scheme> | Protocol supported (HTTP or HTTPS support). Wildcards are supported. |

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<whitelist>
  <uri>
    <host>myhost.example.com</host>
    <port>10201</port>
    <scheme>HTTPS</scheme>
  </uri>
  <uri>
    <host>myhost2.example.com</host>
    <port>6501</port>
    <scheme>HTTP</scheme>
  </uri>
</whitelist>
```

There must be a separate entry for each **host**, **path**, **port** and **scheme** combination. The product now supports whitelisting of URIs with schemes such as t3, t3s, ldap, ldaps, jdbc, etc.

The following components are validated and when:

| Component | Design time validation | Run time validation |
|------------------------------|------------------------|---------------------|
| Message Sender Context | ■ | |
| JNDI Server | ■ | |
| XAI JDBC Connection | ■ | |
| Web Service Adapter | ■ | ■ |
| Feature Configuration | ■ | |
| Reporting Options | ■ | |
| External Navigation Keys | ■ | ■ |
| XAI Executer | ■ | |
| XAI Command | ■ | |
| HTTP Sender | | ■ |
| Real Time HTTP Sender | | ■ |
| JSON Sender | | ■ |
| Real Time Email Sender | | ■ |
| JMS Sender | | ■ |
| SMS BPEL Sender | | ■ |
| Web Services (WSDL, OPA etc) | | ■ |

Support for URI specification using Substitution Variables

Applications need to define URI links to the integrate components for seamless navigation to different application pages, for map viewer, etc. URI specifications can now include substitution variables to allow parameterization of URI components.

This new enhancement featuring substitution variables ensures support for secure, stable and uniformly configured environments. For configuration stability such URL values will be defined using substitution variables, aka aliases. E.g. "@CCB@/demoprop/cis.jsp". This would help in migrating values between different environments, e.g. from CCB Test to CCB Prod environment, and in performing CMA extracts that would allow to pull URL values that are not environment specific.

A substitution list is provided to implement the physical path per environment. This file is maintained in [\\$SPLEBASE/etc/substitutionVariableList.xml](#) (or [%SPLEBASE%\etc\ substitutionVariableList.xml](#) in Windows).

The substitution list file can be specified using the [CLOUD_SUBSTITUTION_VARIABLE_LIST_FILE_LOCATION](#) config parameter in the ENVIRON.INI file.

Following is the format of the substitution variables file:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<substitutionVariables>
  <uriVariable>
    <name></name>
    <value></value>
  </uriVariable>
</substitutionVariables>

```

| Attribute | Comments |
|----------------------------|---|
| <uriVariable> | URI Variable tag |
| <name> | The name of the substitution variable. |
| <value> | The URI(URL/Path) of the substitution variable. |

uriVariable element has the attribute "allURICHildComponentsAreValid" with allowed values being true or false. If the attribute value is set to false, then the uriVariable value needs to be verified against the white list after the variable is replaced with the actual value. If the allURICHildComponentsAreValid is set to true, or is omitted, then this indicates that the variable value is already white listed and there is no further validation of the uri and all its child components (such as path, query and fragement) is needed. Note that the default behaviour is "true".

The URI specification feature automatically includes the variable **SPLOUTPUT**. This is a system variable commonly defined with a valid file path that implementations may write files to. Automatically providing the configuration for this variable provides backward compatibility for any code that supported referencing "@SPLOUTPUT@" for a file location. The SPLOUTPUT variable is automatically identified and substituted without having to explicitly define it in the substitution variable XML file.

For example:

```

<?xml version="1.0" encoding="UTF-8"?>
<substitutionVariables>
  <uriVariable>
    <name>F1_INTERNAL_FILES</name>
    <value>/scratch/OUAF/sploutput/Files/Shared</value>
  </uriVariable>
  <uriVariable>
    <name > F1_CUST_APP_BASE </name>
    <value>/usr/OUAF/CustomerFiles</value>
  </uriVariable>
  <uriVariable >
    <name>CCB</name>

```

```
<value>http://ccbenvurl.oracle.com:8001</value>
</uriVariable>
<substitutionVariables>
```

The following substitution variables are provided by default in OUAF:

1. F1_CMA_FILES
2. F1_BASE_REST_URL
3. F1_BASE_IWS_URI
4. F1_BASE_WEB_URI

Note: The below substitution variables are also provided by default, but no product functionality is using them. In a future release these variables will no longer be delivered (to remove any ambiguity). Implementations are suggested not to use any of the below names, but should rather use CM variable names.

1. F1_BI_EXTRACTS
2. F1_INTERNAL_FILES
3. F1_CUST_APP_BASE
4. F1_PROCESS_DIR
5. F1_SVC_CATALOG_WSDL_DIR
6. F1_PDB_EXTRACTS

*Note: The list **must not** contain variables and file paths that contain internal product infrastructure, e.g. config files, binaries, properties, security info and others.*

*The list **must not** contain existing environment variables such as SPLEBASE, COBOL_RUNTIME_LIB, APPVIEWER_DIR and their corresponding application directories.*

Defining File Alias Using Extendable Lookup

In addition to the above mentioned support for URI specification using substitution variables, an alternate option has been provided that allows for defining native file storage locations using an extendable lookup. The extendable lookup F1-FileStorage (File Storage Configuration) has been provided. Using this extendable lookup, a value may be defined referencing the "Native File Storage" file adapter option and then defining the file path that this value represents. The file path field follows the same rules as any other file path. For example, it can reference "@SPLOUTPUT@" or any other value defined in the Substitution Variables properties file. In addition, if the system has been configured to validate the value against a whitelist, this is also enforced.

A new syntax has been defined for referencing file locations during configuration, for example in "File Path" parameters for a batch control: file-storage://{ExtendableLookupValue}.

Refer to the online help for more information about the supported syntax.

Support for Oracle Cloud Storage Access

The system has been enhanced to allow code that reads or writes files to reference an Oracle Cloud Storage location (container) rather than the native file system. The extendable lookup F1-FileStorage (File Storage Configuration) has been provided to support this. Using this extendable lookup, a value may be defined referencing the "Oracle Cloud Object Storage" file adapter option. With that configuration, your implementation may define the REST Endpoint URL, user name and password for accessing the storage location. Using this configuration, the same syntax described above may then be used when defining a File Path in the application, for example a parameter for a batch control.

```
file-storage://{ExtendableLookupValue}/{serviceName}
```

Refer to the online help for more information about the supported syntax.

Consolidated logging

Note: This feature only applies to clustered environment of the product.

Utilities products provisioned on the cloud would be hosted across multiple cluster nodes. The Application server or the Inbound Web Services could be spread across multiple nodes making their corresponding logs, distributed. In order to debug or to monitor the application components, users are provided with a consolidated view of these distributed logs.

Oracle Utilities Application Framework leverages RSysLog to accomplish the task to consolidate the distributed logs.

Note: The configuration and usage of RSysLog is not covered in this document.

Users can configure the location of the distributed logs using the **'com.oracle.ouaf.consolidatedLog.fileName'** parameter in the spl.properties file. If configured, the consolidated logs would be picked up from the specified logs file and displayed to the user.

The RSysLog should be configured to retrieve the distributed logs and consolidate them into the log file specified against the **com.oracle.ouaf.consolidatedLog.fileName** property.

Domain Template Support

Note: This feature only applies to Oracle WebLogic 12 and above implementations only.

Note: Before using Domain Templates refer to the [Fusion Middleware Creating Domain Templates Using the Domain Template Builder](#) for more information about Domain Templates and Extension Templates.

Oracle WebLogic supports the concept of [Domain Templates](#). These are predefined domain configurations that can be used to establish the domain configuration reducing the need for manual effort in building domains.

The product now ships with a number of base templates that can be used when creating the domain:

- **oracle-utilities-simple-linux-<version>.jar** - This is a template useful

for non-production installations with an Administration server and single product server with the base domain attributes, as per the native installation.

- **oracle-utilities-complex-linux-<version>.jar** - This is a template useful for clustered production installations with an Administration server and single product cluster (with a single server) with the base domain attributes, as per the native installation.

These templates are located in **\$SPLEBASE/tools/domaintemplates**. These can be used with the [Oracle WebLogic Configuration Wizard](#) or Oracle WebLogic installer.

If a native mode installation is desired, it is recommended to use the domain templates as provided to reduce the amount of manual configuration.

Oracle WebLogic ships with a [Template Builder](#) to allow implementations to build their own custom templates. If the domain has been customized for any reason, then using a custom domain template can simplify the provisioning of new domains.

When using the domain template builder ensure the following is included in your custom domain template:

- Ensure the servers allocated to the product are included.
- Ensure all the deployments allocated to the product are included.
- Ensure the secure domain configuration is included. Optionally users can also be included. At a minimum the administration users should be transferred, if desired.
- Ensure the XML registry is included.
- Ensure JDBC and JMS resources used by the product are also included.

Diagnostics Support (WLDF)

Note: This feature only applies to Oracle WebLogic 12 and above implementations only.

Note: Refer to [Configuring and Capturing Diagnostic Images](#) for additional advice.

Oracle WebLogic includes a diagnostics capability in the form of [Oracle WebLogic Diagnostics Framework](#) (WLDF). Typically most diagnostics tools only detect an issue after it has happened, which means important diagnostic information may not be captured. The WLDF captures information at the time of the issue and after to provide additional diagnostics in a standard format for inclusion in diagnostic tools built into Oracle products and for Service Requests within My Oracle Support.

The diagnostic engines works in the following way:

- The diagnostics facility is enabled to capture information. As part of that enablement the location of the diagnostics is configured.
- A set of rules are added to configure the diagnostics engine to recognize issues that are above and beyond the typical issues. WLDF detects common java issues automatically.
- When a rule or the diagnostic engine detects an error is occurring or has occurred, it gathers as much diagnostics information it can at the time of the error and creates a diagnostic image which assembles the diagnostics information into a structured format. This is placed in the configured location.
- Diagnostic images are available from the console as well as Oracle Enterprise Manager where they can be viewed, analyzed and attached to Service Requests as necessary.

The product now supports the WLDF using the following:

- A set of configuration settings that enable the support as well as set some basic settings. These settings are primarily used for embedded installations. For native installations, the WLDF definitions are included in the provided [Domain Templates](#).

Note: For customers with preexisting native installations wanting to use this facility, it is recommended to enable the facility in an embedded installation and manually make the changes to your domain to match the settings using the Oracle WebLogic console or Oracle Fusion Middleware console.

- A set of base rules to instruct WLDF the conditions to track and detect issues. These are customizable rules for Oracle WebLogic to detect any issues outside of typical issues.

The following parameters control the implementation of the Diagnostics Framework:

- [WLS_DIAGNOSTIC_CONTEXT_ENABLED](#) - This enables (**true**) or disables (**false**) the diagnostics generation support in the product.
- [WLS_DIAGNOSTIC_BUILT_IN_MODULE](#) - If [WLS_DIAGNOSTIC_CONTEXT_ENABLED](#) is enabled, this sets the name of the diagnostics module to be used for diagnostics. The default value is **None**.
- [WLS_DIAGNOSTIC_VOLUME](#) - If [WLS_DIAGNOSTIC_CONTEXT_ENABLED](#) is enabled, this determines the volume of diagnostics generated for the diagnostic tools such as [Flight Recorder](#). The default is **off**.

For native installations, the following process should be used from the Oracle WebLogic console to enable support:

- Navigate to the *Environment* → *Servers* menu item and click on the product server name to enable.
- In the *Configuration* tab, change the value of *Diagnostic Volume* to the desired level of diagnostics. It is recommended to use **LOW** as a minimum. *Save* the changes and repeat for each product server (if multiple are implemented in the domain).
- Navigate to the *Diagnostics* → *Built-in Diagnostics Modules* menu item and select the product servers. Click on the *Activate* button to enable basic diagnostics. This should change the *Status* to **Active**.
- If a custom Diagnostic Module is to be used, then navigate to the *Diagnostics* → *Diagnostic Modules* menu item and select the desired diagnostic module to maintain. From the *Targets* tab, select the product servers to apply the desired diagnostics module. *Save* the changes.

ECID Support

Note: This feature only applies to Oracle WebLogic 12 and above implementations only.

Oracle WebLogic includes a set of facilities to add additional information to transactions known as the [Execution Context Identifier \(ECID\)](#). The single ECID is attached to each unique transaction within the product. This allows diagnostic tools to perform forensic analysis based on the identifier across the architecture.

There are a set of rules that apply to the ECID within the product:

- A unique ECID is generated at the start of a transaction within the product. This ECID

is active for all calls within the product till a **COMMIT** or **ROLLBACK** is issued by the product. This constitutes a transaction boundary within the product.

- The ECID is only generated for online web based transactions in the present release.
- The ECID is used within the logs, diagnostic images and transaction tracking facilities within Oracle WebLogic and the related database connections²¹.
- The use of ECID is optional and disabled, by default, for backward compatibility.
- The ECID is used by the various Oracle diagnostics tools such as [Oracle Enterprise Manager](#), [JVMD](#) and [Business Transaction Management](#).

The following parameters control the implementation of the Diagnostics Framework:

- [WLS_DIAGNOSTIC_CONTEXT_ENABLED](#) - This enables (**true**) or disables (**false**) the diagnostics generation support in the product.

For native installations, the following process should be used from the Oracle WebLogic console to enable support:

- Navigate to the *Diagnostics* → *Context* menu item.
- Select the product servers and click on *Enable* to enable ECID support.
- The status of the selected servers should change to *Enabled*.

Global Cache Flush Support

Note: This facility is only available in Oracle WebLogic.

One of the architectural features of the products is the support for caching of commonly used data. This cache is used to prevent the transactions to perform unnecessary calls for commonly used data. This cache is loaded and maintained by the product automatically.

In some cases, it may be necessary to force a refresh of the cache to reload the static data. Whilst a number of utilities are provided to [flush the online/web services](#) cache as well as the [batch cache](#), if there were a number of servers or threadpools then the cache request may have to be performed on each threadpool or server.

In the current release any cache request can be configured to flush all caches, known as a global flush, across the architecture. Use of this facility is optional, but recommended in production environments to ensure consistent processing.

To use this facility the following must be configured:

- The [ouaf.flush.jms.disabled](#) property in the [spl.properties](#) file must be set to **false**.
- The [ouaf.flush.jms.connection](#), [ouaf.flush.jms.requestTopic](#) and [ouaf.flush.jms.responseTopic](#) properties must be populated in the [spl.properties](#) with the JMS resources used. These JMS resources must be created within the domain of the product.

Once enabled all flush requests now are global.

The global flush operates in the following way:

- When a flush is requested from the online using the JSP or JMX facility, Web Services

²¹ This is only supported when using Oracle WebLogic JDBC Data Sources. It is not supported for UCP at the present time.

via JMX and batch via running **F1-FLUSH**, an entry is placed in the JMS Topic dictated by [ouaf.flush.jms.requestTopic](#).

- Each of the components of the architecture has subscribed to the JMS resource specified by [ouaf.flush.jms.requestTopic](#) at their startup time.
- When they receive a message in that topic, each component flushes their cache according to their local methods and sends an appropriate response in the JMS resource specified by [ouaf.flush.jms.responseTopic](#) indicating success or failure. This JMS resource may be monitored using the Oracle WebLogic console, Oracle Enterprise Manager, a JMS Browser or the JMS Viewing facility within the product.

Note: It is possible to configure additional features of the JMS resources outlined above to decide on retention periods and other settings. Refer to the [Understanding Oracle WebLogic JMS](#) documentation for additional features.

Java Flight Recorder Support

Note: This feature only applies to Oracle JDK 7u60 and above implementations only.

Note: If Java Flight Recorder is enabled then these will also be included in Diagnostic Images if [Diagnostics](#) is enabled.

The [Java Flight Recorder](#) is a component of Java that collected diagnostic and profile data for a running Java Virtual Machine (JVM). The product now supports the enabling of the Java Flight Recorder for tracking online java performance.

To enable the support for Java Flight Recorder in a native installation the following process must be performed on the Oracle WebLogic console:

- Navigate to the *Environment* → *Servers* menu item. Click on the product server to enable configuration.
- Navigate to the *Server Start* tab on the *Configuration* tab and add the following java options to the *Arguments* field.
`-XX:+UnlockCommercialFeatures -XX:+FlightRecorder`

Note: If your site is using the [setUserOverrides.sh](#) facility, place these arguments in that script as an alternative.

- Navigate to the *General* tab on the on the *Configuration* tab and alter the *Diagnostics Volume* to the desired amount of diagnostics to capture. At a minimum this value should be set to **LOW**.
- Optionally, enable [ECID tracking](#) for advanced diagnostics for tracing individual transactions.
- Edit any Diagnostic Modules you are using, or create one if you are not using a module and configure a DyeInjection Monitor as outlined in [Configuring the DyeInjection Monitor to Manage Diagnostic Contexts](#). Filters may be specific to focus the scope of the monitor if desired.

Work Manager Support

Note: This feature only applies to native installation implementations only.

Note: This feature is only available in Oracle WebLogic implementations only.

Oracle WebLogic supports [Work Managers](#) that allow implementations to set limits on the server to help optimize performance and prevent server overloading. The product now allows specification of Work Managers to provide additional throttling capabilities.

To use this feature the following must be performed within the Oracle WebLogic console:

- Create a Work Manager with an appropriate name using the *Environment* → *Work Managers* menu item. Target the Work Manager to the product servers using the *Targets* tab. By default, the Work Manager will be unlimited and have no restrictions.
- The product supports the following [constraints](#):
 - Minimum Threads Constraint
 - Maximum Threads Constraint
 - Capacity Constraint
- It is also possible to override the action used for Stuck Threads.
- Request Class is currently not supported.
- *Save* the changes.

Work Managers can be monitored using Oracle WebLogic console or Oracle Enterprise Manager. Refer to the [Work Managers](#) documentation for additional information.

Trust Store Support

Note: For more information about trust stores refer to the [JSSE documentation](#) and [Keystore Configuration](#) (Doc Id: [2014161.1](#)).

The product now supports trust stores as well as key stores for two phase authentication for the online web channel. The format of a trust store is the same as the keystore but the trust store is used to verify credentials whilst the keystore provides the credentials.

In order to configure the truststore the configuration settings in the custom version of `spl.properties.keystore.truststore.include` user exit file needs to be provided.

Miscellaneous Operations And Configuration

Enabling Email Logging from Log4j2

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 <https://logging.apache.org/log4j/2.x/manual/appenders.html#SMTPAppender> for details of the appender.

The following changes are required to enable this.

- 1) Open the log4j2.properties in the relevant **\$SPLEBASE/etc/conf** (**%SPLEBASE%\etc\conf** in Windows) subdirectory:

- Web Application Server – [log4j2.properties](#)
- Business Application Server - [log4j2.properties](#)

- 2) Add the following lines to the file:

```
### E1 is an EmailAppender
log4j2.appender.E1 = org.apache.log4j2.net.SMTPAppender
log4j2.appender.E1.Threshold = ERROR
log4j2.appender.E1.layout = org.apache.log4j2.PatternLayout
log4j2.appender.E1.layout.ConversionPattern = %d{ISO8601} [%t] %-5p
%C %x - %m%n
log4j2.appender.E1.From = <from>
log4j2.appender.E1.SMTPHost = <SMTPHost>
log4j2.appender.E1.Subject = <subject>
log4j2.appender.E1.To = <to>
###
### The following settings are optional
###
log4j2.appender.E1.SMTPUsername = <SMTPUsername>
log4j2.appender.E1.SMTPPassword = <SMTPPassword>
log4j2.appender.E1.CC = <cc>
log4j2.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 |

| Parameter | Field from example | Usage |
|--------------|-----------------------------------|---|
| BCC | <code><bcc></code> | Email address/group to send "blind" courtesy copy of emails to |
| SMTPHost | <code><SMTPHost></code> | Host Name of SMTP Server |
| SMTPUsername | <code><SMTPUsername></code> | Logon User for SMTP Server (<i>if supported</i>) |
| SMTPPassword | <code><SMTPPassword></code> | Password for Logon User for SMTP Server (<i>if supported</i>) |
| Subject | <code><subject></code> | 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' ###
log4j2.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

| | |
|-----------------------------------|----------------------------|
| <code><hostname></code> | Database hostname |
| <code><dbport></code> | Database Listener portname |
| <code><database_sid></code> | 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, `sp1sys`).
- 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.*

Centralized Properties Customization

Support is now added for a centralized properties customization file using which properties present in various configuration files can be customized. Instead of customizing individual properties files, a single file can be used to specify all the properties that are to be customized (add, modify or remove).

Once specified in this custom central properties file, the properties are automatically migrated to specified properties files. All the configuration properties should be specified in the custom centralized properties file "**etc/cm_properties.ini**" (this can be created as a customization). If the file exists the content will be processed for the relevant properties.

Each entry in the centralized properties customization file should be specified in the following format:

```
<PROPERTIES_FILE> : <PROPERTY_NAME> = <VALUE>
```

| Parameter | Usage |
|-----------------|---|
| PROPERTIES_FILE | The property file in which the property |

| Parameter | Usage |
|---------------|--|
| | needs to be added/updated/removed. |
| PROPERTY_NAME | The name of the property. |
| VALUE | The value to be added/updated against the property in the specified property file. |

- For each property entry(<PROPERTIES_FILE> : <PROPERTY_NAME>=<VALUE>) specified in the centralized properties customization file, the property(PROPERTY_NAME=VALUE) is added in the specified properties file(PROPERTIES_FILE), if doesn't exist in the specified property file.
- If the property(PROPERTY_NAME=VALUE) already exists in the specified properties file(PROPERTIES_FILE), then the entry is updated with the specified value(VALUE).
- If user doesn't specify the properties file name in the entry i.e., the entry in the centralized properties customization file is "<PROPERTY_NAME>=<VALUE>", then the <PROPERTY_NAME> in all property files is overridden with <VALUE>, if <PROPERTY_NAME> exists.
- In order to remove an existing property from a specific properties file, the centralized properties file will need to have the entry specified as "<PROPERTIES_FILE> : <PROPERTY_NAME>=[DELETE]"
- In order to remove an existing property from a all the properties file, the centralized properties file will need to have the entry specified as "<PROPERTY_NAME>=[DELETE]"

Sample entry in the centralized properties customization file:

```
hibernate.service.properties.template:hibernate.user=clouduser
```

This sets the hibernate.user property to clouduser in the hibernate.service.properties file.

Note: This feature is implemented if the [initialSetup.sh](#) script is executed with the minimum of -t option.

Cache Management

Note: If cache management requires global cache management across various nodes then refer to the [Global Cache Flush Support](#) section for more details.

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 | <code>flushAll.jsp</code> | <code>flushAll</code> |
| Refresh all drop down data | <code>flushDropdownCache.jsp</code> | <code>flushDropDownCache</code> |
| Refresh field labels | <code>flushMessageCatalog.jsp</code> | <code>flushMessageCatalog</code> |
| Refresh Fields and FK information | <code>flushFieldAndFKMetaData.jsp</code> | <code>flushFieldAndFKMetaData</code> |
| Refresh menu items | <code>flushMenu.jsp</code> | <code>flushMenu</code> |
| Refresh messages | <code>flushMessaging.jsp</code> | <code>flushMessaging</code> |
| Refresh navigation keys | <code>flushNavigationInfo.jsp</code> | <code>flushNavigationInfo</code> |

| Function | JSP | MBean Operation |
|---------------------------------|---------------------------------------|-----------------------------------|
| Refresh portals and zones | <code>flushPortalMetaInfo.jsp</code> | <code>flushPortalMetaInfo</code> |
| Refresh screen style sheets | <code>flushUI_XSLs.jsp</code> | <code>flushUIXSLs</code> |
| Refresh security | <code>flushSystemLoginInfo.jsp</code> | <code>flushSystemLoginInfo</code> |
| Refresh specific drop down data | <code>flushDropDownField.jsp</code> | <code>flushDropDownField</code> |

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 | <ul style="list-style-type: none"> • Select <i>Tools</i> on your browser menu bar • Select <i>Internet Options</i> on the menu that appears. • Click the <i>Delete Files</i> button on the pop-up that appears. • Click the <i>Delete all...</i> button on the subsequent pop-up that appears and then click OK. • Enter the standard product URL to re-invoke the product. |
| Mozilla Firefox | <ul style="list-style-type: none"> • Select <i>Tools</i> from your browser menu bar. • Click <i>Options</i> on the Tools menu. • Select the <i>Advanced</i> tab from the Options dialog. • Select the <i>Network</i> tab from the <i>Advanced</i> tab. • Click on the <i>Clear Now</i> 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 this parameter to 1 second on development / test environments and 28800 seconds (8 hours) on production environments.

Oracle WebLogic: Expanded or Archive Format

By default, the product is built into a set of WAR/EAR files and deployed in this format on Oracle WebLogic 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 JEE 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:

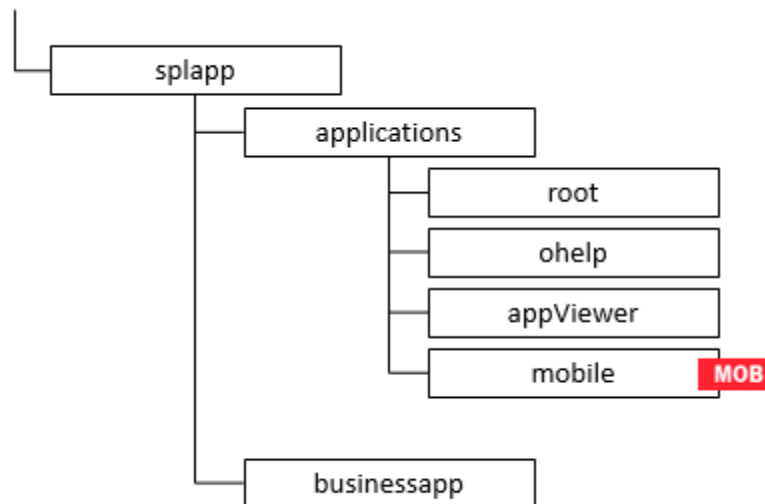


Figure 39 – Expanded Format

- Under the **root** directory are the product specific subdirectories for each subsystem or part of the online component of the product.
- Under the **businessapp** directory are the business object specific files for each subsystem or part of the online component of the product.
- The **ohelp** and **Appviewer** directories contain an expanded mode version of the **ohelp** and **appViewer** generated files (after running [genappvieweritems](#)).
- Under the **mobile** directory are the Mobile Platform specific files (if the mobile framework is deployed with your product). **MOB**

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 product 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 be 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 IWS components:

| Templates | Configuration File | Usage |
|--|---|---|
| application.xml.template | applicaton_web.xml | JEE global application configuration file, which contains common settings for the Web Application Server |
| application_service.xml.template | application_service.xml | JEE 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 |

| Templates | Configuration File | Usage |
|---|---|---|
| | | boot credentials file used for starting server |
| <code>coherence-cache-config.xml.template</code> | <code>coherence-cache-config.xml</code> | Batch Coherence cache settings. BATCH |
| <code>doc1dirfile.ini.template</code> | <code>doc1dirfile.ini</code> | Bill Print extract configuration file |
| <code>e0Batch.properties.template</code> | <code>e0Batch.properties</code> | Global batch properties BATCH |
| <code>earServiceBuild.xml.template</code> | <code>earServiceBuild.xml</code> | ANT Build file for EAR file for Business Application Server |
| <code>earwebBuild.xml.template</code> | <code>earwebBuild.xml</code> | ANT Build file for EAR file for Web Application Server |
| <code>ejb-jar.xml.template</code> | <code>ejb-jar.xml</code> | Generic Business Application Server descriptor for EJB's |
| <code>Fault.xsd.iws.template</code> | <code>Fault.xsd</code> | Web Services Fault Schema |
| <code>iws-build-common.xml.iws.template</code> | <code>iws-build.xml</code> | Common settings for IWS Build |
| <code>iws-build-WLS.xml.iws.template</code> | <code>iws-build.xml</code> | IWS build Settings for Oracle WebLogic |
| <code>iwsdeploy-log4j2.properties.iws.template</code> | <code>log4j2.properties</code> | Log properties for IWS build process |
| <code>jarservice.xml.template</code> | <code>jarservice.xml</code> | ANT Build file for jar files. |
| <code>java.login.config.template</code> | <code>java.login.config</code> | JAAS Login file |
| <code>log4j2.properties.iws.template</code> | <code>log4j2.properties</code> | Log configuration for IWS |
| <code>log4j2.properties.service.template</code> | <code>log4j2.properties</code> | Log configuration for BAS |
| <code>log4j2.properties.standalone.template</code> | <code>log4j2.properties</code> | Log configuration |

| Templates | Configuration File | Usage |
|--|---|--|
| | | for batch |
| log4j2.properties.template | log4j2.properties | Log configuration for online |
| 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 Doc Id: 970785.1 to My Oracle Support for details of this integration. |
| 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 . |

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.

Sending emails at the conclusion of batch process **BATCH**

It is possible to send a notification email when a batch process has ended. This notification happens after the batch process has ended and all application-related commits/rollbacks have

taken place. It does not impact the batch process itself in the event of errors happening during the notification process. The default email is a simple text email that contains the batch control, date and time of the submission, run number, submission parameters, batch process summary indicating records processed and in-error, as well as the thread details, including logged messages (up to 100).

The email address can be an individual person or a valid mail group (the latter requires additional configuration in your email system).

To use email notification the email server must be configured using one of the following options:

1. The mail server can be defined through the default Message Sender (Class **RTEMAILSND**R) with the appropriate SMTP settings on the Context tab.

2. Alternatively the properties can be supplied in the form of JVM properties as follows:

```
# Host whose mail services will be used
# (Default value : localhost)
mail.host=<your mail server>
# Return address to appear on emails
# (Default value : username@host)
mail.from=<name@host>
```

```
# Other possible items include:
# mail.user=
# mail.store.protocol=
# mail.transport.protocol=
# mail.smtp.host=
# mail.smtp.user=
# mail.debug=
```

| Name | Type | Description |
|---------------------------------------|---------|---|
| <code>mail.debug</code> | boolean | The initial debug mode. Default is false. |
| <code>mail.from</code> | String | The return email address of the current user, used by the InternetAddress method getLocalAddress . To ensure that the value of <code>mail.from</code> property is handled properly when set through a user exit include file, add a backslash (\) in front of the '@' symbol. e.g. If the value desired for <code>mail.from</code> is <code>demo@orgname.org</code> then it should be set as <code>demo\@orgname.org</code> |
| <code>mail.host</code> | String | The default host name of the mail server for both Stores and Transports. Used if the mail.protocol.host property isn't set. |
| <code>mail.mime.address.strict</code> | boolean | The <code>MimeMessage</code> class uses the <code>InternetAddress</code> method parseHeader to parse headers in messages. This property controls the strict flag passed to the |

| Name | Type | Description |
|--------------------------------------|--------|--|
| | | parseHeader method. The default is true. |
| <code>mail.smtp.class</code> | String | Specifies the fully qualified class name of the provider for the specified protocol. Used in cases where more than one provider for a given protocol exists; this property can be used to specify which provider to use by default. The provider must still be listed in a configuration file. |
| <code>mail.smtp.host</code> | String | The host name of the mail server for the specified protocol. Overrides the mail.host property. |
| <code>mail.smtp.port</code> | int | The port number of the mail server for the specified protocol. If not specified the protocol's default port number is used. |
| <code>mail.smtp.user</code> | String | The user name to use when connecting to mail servers using the specified protocol. Overrides the mail.user property. |
| <code>mail.store.protocol</code> | String | Specifies the default message access protocol. The Session method getStore() returns a Store object that implements this protocol. By default the first Store provider in the configuration files is returned. |
| <code>mail.transport.protocol</code> | String | Specifies the default message access protocol. The Session method getTransport() returns a Transport object that implements this protocol. By default the first Transport provider in the configuration files is returned. |
| <code>mail.user</code> | String | The default user name to use when connecting to the mail server. Used if the mail.protocol.user property isn't set. |

These properties can be added to the **threadpoolworker.properties** file for the standalone batch **threadpoolworker**, or the **spl.properties** file for an online application server that hosts a batch worker.

Oracle WebLogic Configuration Support

The product has *native* support for Oracle WebLogic. Normally the JEE Web Application is installed and the JEE 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 JEE Web Application Server directories (varies according to JEE 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 JEE 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 Webservices) | web.xml | ✗ |
| | spl.properties | ✓ |
| | weblogic.xml | ✗ |

| Component | Configuration File | Externalized |
|-----------------------------|--------------------------------------|--------------|
| Business Application Server | log4j2.properties | ✓ |
| | web.xml | ✗ |
| | spl.properties | ✓ |
| | hibernate.properties | ✓ |
| | log4j2.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.onserver** 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:

https://<host>:<port>/<server>/cm/privacy.html

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

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:

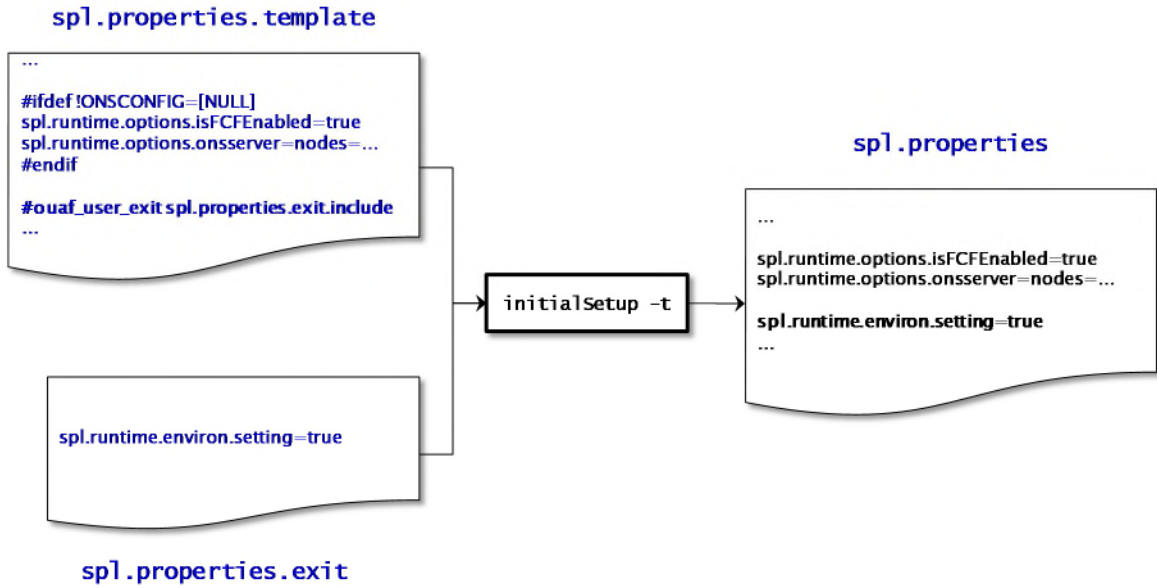


Figure 40 – Include files

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:

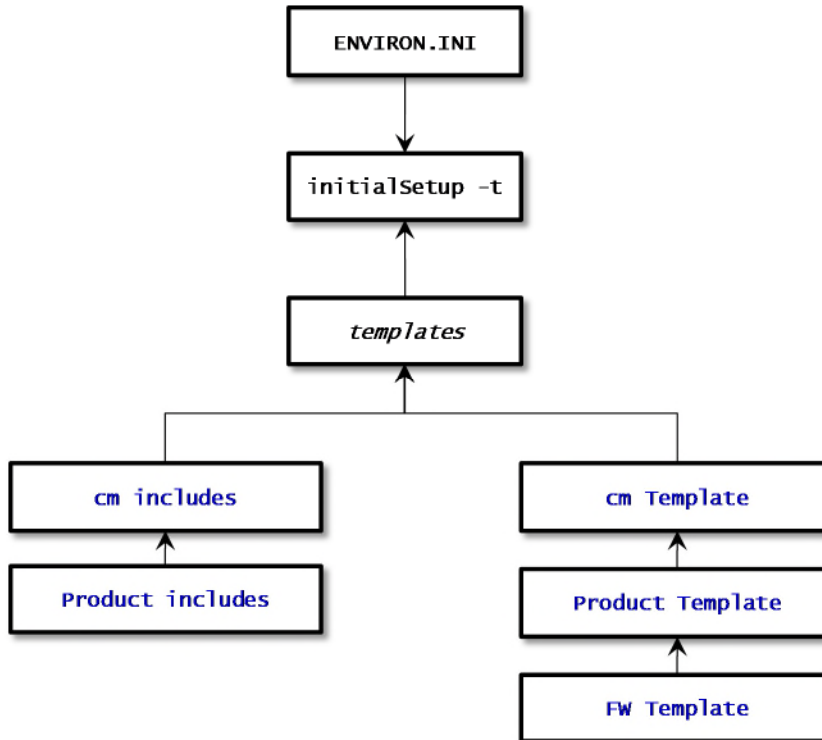


Figure 41 – Template Hierarchy

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

| Template File | User Exit Include file | Position and Usage |
|---------------------------------------|---|--|
| <code>boot.properties.template</code> | <code>boot.properties.exit.include</code> | Used to change boot properties file from Oracle WebLogic |

| Template File | User Exit Include file | Position and Usage |
|---|---|--|
| <code>ejb-jar.xml.template</code> | <code>ejb-jar.xml.wls.jms_1.include</code> | JMS Mappings for Oracle WebLogic |
| <code>hibernate.properties.web.template</code> | <code>hibernate.properties.exit.include</code> | At end of file (common hibernate.properties entries) |
| | <code>hibernate.properties.web.exit.include</code> | At end of file (online specific hibernate.properties entries) |
| <code>log4j2.properties.template</code> | <code>log4j2.properties.exit.include</code> | At end of file (common log4j2.properties entries) |
| | <code>log4j2.properties.root.exit.include</code> | At end of file (specific online log4j2.properties entries) |
| <code>log4j2.properties.service.template</code> | <code>log4j2.properties.exit.include</code> | At end of file (common log4j2.properties entries) |
| | <code>log4j2.properties.service.exit.include</code> | At end of file (specific XAI log4j2.properties entries) |
| <code>ouaf.jmx.access.file.template</code> | <code>ouaf.jmx.access.file.exit.include</code> | Allows for additional users to be specified for JMX connections |
| <code>ouaf.jmx.password.file.template</code> | <code>ouaf.jmx.password.file.exit.include</code> | Allows for additional passwords to be specified for JMX users |
| <code>sp1.properties.service.template</code> | <code>sp1.properties.exit.include</code> | At end of file (common sp1.properties entries) |
| | <code>sp1.properties.service.exit.include</code> | At end of file for EJB sp1.properties entries. |
| | <code>sp1.properties.service.timeouts.exit.include</code> | User exit for service timeouts. |
| <code>sp1.properties.template</code> | <code>sp1.properties.exit.include</code> | At end of file (common sp1.properties entries) |
| | <code>sp1.properties.root.exit.include</code> | At end of file for Web Application based sp1.properties entries. |
| | <code>sp1.properties.timeouts.root.exit.include</code> | User exit for global timeouts |

| Template File | User Exit Include file | Position and Usage |
|---|--|---|
| | <code>FW_sp1.properties.keystore.truststore.include</code> | User exit for keystore and trust store settings |
| <code>sp1.properties.mobile.template</code> | <code>sp1.properties.exit.include</code> | At end of file (common sp1.properties entries) |
| | <code>sp1.properties.mobile.exit.include</code> | At end of file for mobile Application based sp1.properties entries. |
| | <code>sp1.properties.timeouts.mobile.exit.include</code> | Future use |
| <code>submitbatch.properties.template</code> | <code>submitbatch.properties.exit.include</code> | Submitter extensions |
| <code>threadpoolworker.properties.template</code> | <code>threadpoolworker.properties.exit.include</code> | Threadpool extensions |
| <code>web.xml.template</code> | <code>sp1.properties.images.include</code> | Image processing overrides for web.xml |
| | <code>web.xml.servlet_mapping.include</code> | Allow custom servlet mappings |
| | <code>web.xml.servlet.include</code> | Allow custom servlet definitions |
| | <code>sp1.properties.filter_mapping.include</code> | 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 |
|-----------------------------------|--|
| <code>hibernate.properties</code> | <code>hibernate.properties.exit.include</code> |
| <code>log4j2.properties</code> | <code>log4j2.properties.exit.include</code> |
| <code>sp1.properties</code> | <code>sp1.properties.exit.include</code> |

- **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 | <code>hibernate.properties</code> | <code>hibernate.properties.web.exit.include</code> |
| | <code>log4j2.properties</code> | <code>log4j2.properties.root.exit.include</code> |
| | <code>sp1.properties</code> | <code>sp1.properties.root.exit.include</code> |
| Business App Server | <code>log4j2.properties</code> | <code>log4j2.properties.service.exit.include</code> |
| | <code>sp1.properties</code> | <code>sp1.properties.service.exit.include</code> <code>sp1.properties.service.timeouts.exit.include</code> |
| Inbound Web Services | <code>log4j2.properties</code> | No User Exit available |
| | <code>sp1.properties</code> | No User Exit available |
| Batch Server BATCH | <code>hibernate.properties</code> | <code>hibernate.properties.batch.exit.include</code> |
| | <code>log4j2.properties</code> | <code>log4j2.properties.standalone.exit.include</code> |
| | <code>sp1.properties</code> | <code>sp1.properties.standalone.exit.include</code> <code>sp1.properties.timeouts.standalone.exit.include</code> |

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 JEE Web Application Server²². 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 JEE 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

²² It is possible to define the physical JMS configuration using the console provided with the JEE 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:

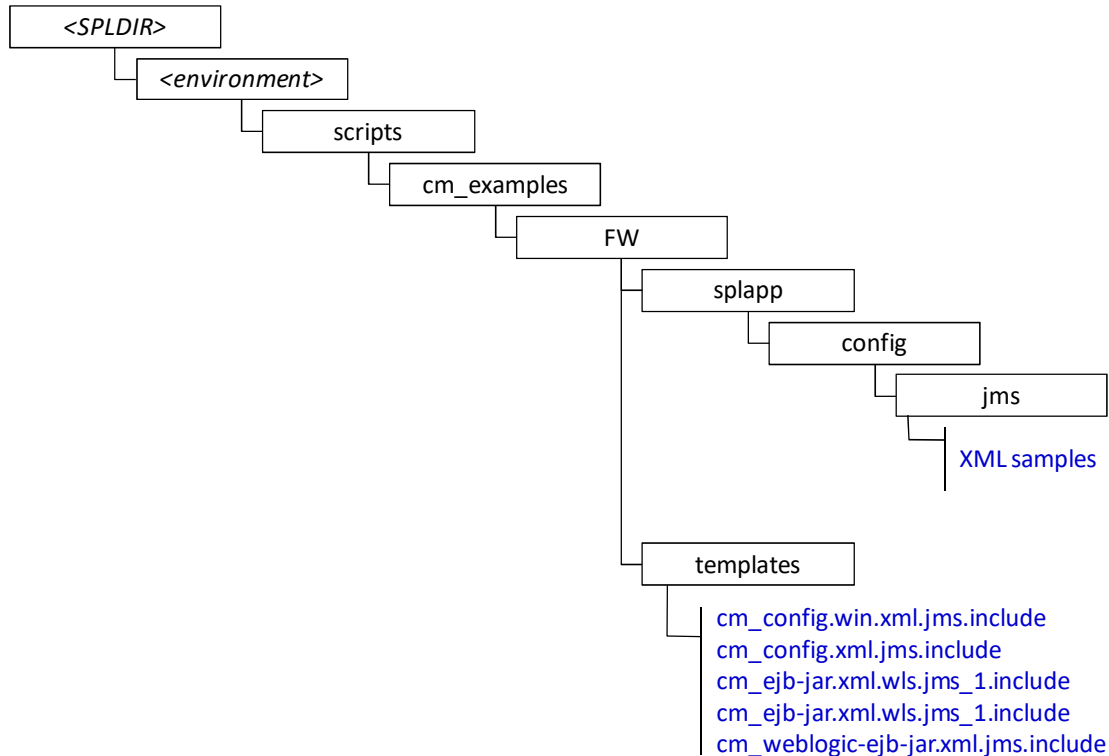


Figure 42 – Custom JMS configuration

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) | <ul style="list-style-type: none"> • 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) | <ul style="list-style-type: none"> • 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 ouaf.timeout.script.default parameter to the desired timeout (in seconds) to set a global default on service script invocations. • Set the |

| Tier/configuration file | Configuration Setting |
|-------------------------|---|
| | <code>ouaf.timeout.service.default</code> 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: <code>ouaf.timeout.business_object.default</code> Specific: <code>ouaf.timeout.business_object.<bocode></code> |
| Business Services | Default: <code>ouaf.timeout.business_service.default</code> Specific: <code>ouaf.timeout.business_service.<bocode></code> |
| Query Zones | Default: <code>ouaf.timeout.query.default</code> Specific: <code>ouaf.timeout.query.<zonename></code> |
| Scripts | Default: <code>ouaf.timeout.script.default</code> Specific: <code>ouaf.timeout.script.<scriptname></code> |
| Application Services | Default: <code>ouaf.timeout.service.default</code> Specific: <code>ouaf.timeout.service.<service></code> |

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 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) and/or Batch (standalone).
- To set the feature at the global level for an environment, navigate to the *Administration* menu → *General* → *Feature Configuration Add* menu option and add a *General System Configuration* Feature Type with the *System Override Date* option in **YYYY-MM-DD** format. For example:

| Feature Configuration | | | |
|-----------------------|------------------------------|----------|----------------------|
| Main | | Messages | |
| FEATURE NAME | <input type="text"/> | | |
| FEATURE TYPE | General System Configuration | | |
| DESCRIPTION | <input type="text"/> | | |
| OPTIONS | | | |
| | OPTION TYPE | SEQUENCE | VALUE |
| + | System Override Date | 1 | <input type="text"/> |

Figure 43 – Feature Configuration

- 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 → *Security* → *User Search* menu option) with the Characteristic Value set to the desired date in **YYYY-MM-DD** format. For example:

| User | | | | | | | | | | | | | | | |
|---------|----------------------|----------------------|----------------------|-----------------|-----------|--------------------|--|-----------|--|----------------|--|------------------|--|-----------------|--|
| Main | | To Do Roles | | Access Security | | Portal Preferences | | Bookmarks | | Favorite Links | | Favorite Scripts | | Characteristics | |
| USER ID | SYSUSER | | | OWNER | Framework | | | | | | | | | | |
| | CHARACTERISTIC TYPE | SEQUENCE | CHARACTERISTIC VALUE | | | | | | | | | | | | |
| + | System Override Date | <input type="text"/> | <input type="text"/> | | | | | | | | | | | | |

Figure 44 – System override for individual users

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.

Secure Transactions

Note: Refer to the Security Guide for the product for an indepth discussion on implementing SSL.

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.

Native Oracle WebLogic Support

Note: The product also now supports the Oracle Fusion Middleware console as an alternative to the Oracle WebLogic console. This manual covers the Oracle WebLogic console only. Customers using the Fusion Middleware console should use the corresponding feature in that console.

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 install multiple Oracle WebLogic installation and use Oracle Enterprise Manager to manage the multiple instances.

Note: The product now includes a set of base [domain templates](#) that can be used with the

installation of the product to avoid manual installation issues. [Domain Template Builder](#) can also be used to build custom domain templates.

- 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.

Manual Re-Deployment Process

Whenever the installation EAR/WAR files are updated they must be redeployed to the JEE Web Application Server to implement the change using one of the following techniques:

- Using the Oracle WebLogic console, use the *Update* transaction on the selected *Deployments*. The wizard will remember the last settings, which need to be confirmed, prior to deployment. For example:

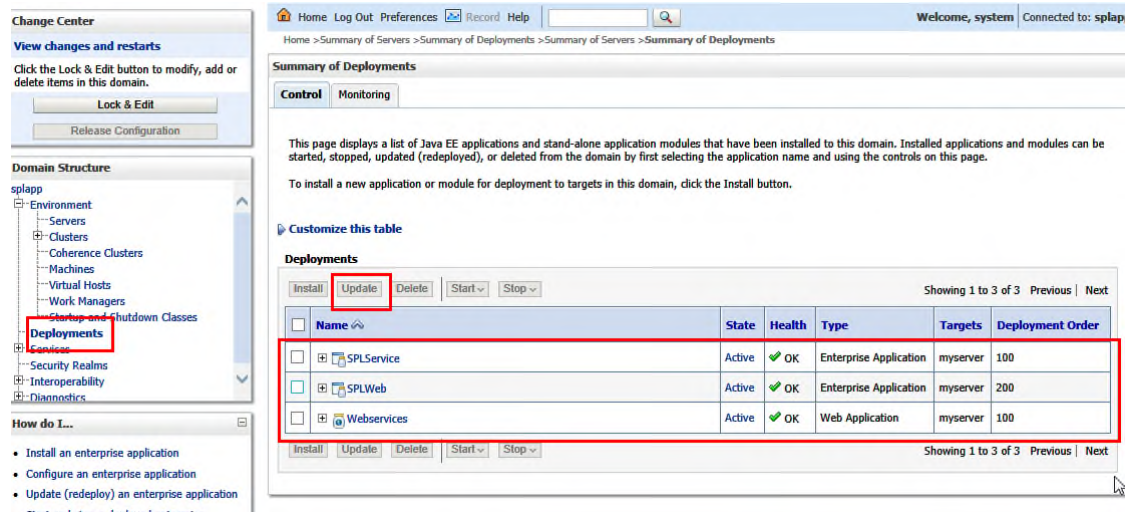


Figure 45 – Update Oracle WebLogic Deployment

- Use the WLST command line using the [redeploy](#) command in the Deployment API. Refer to the [WLST Reference](#) for more information.

The following situations require a redeployment event:

- Any execution of [initialSetup](#) or [initialSetup -w](#) which requires [SPLService](#) (or the value of [SPLSERVICEAPP](#)) and [SPLWeb](#) (or the value of [SPLWEBAPP](#)) to be redeployed.
- Any execution of [iwsdeploy.sh](#) or online deployment of Inbound Webservices. This requires Webservices (or the value of [IWSWAR](#)) deployment

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 account²³.
- 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:

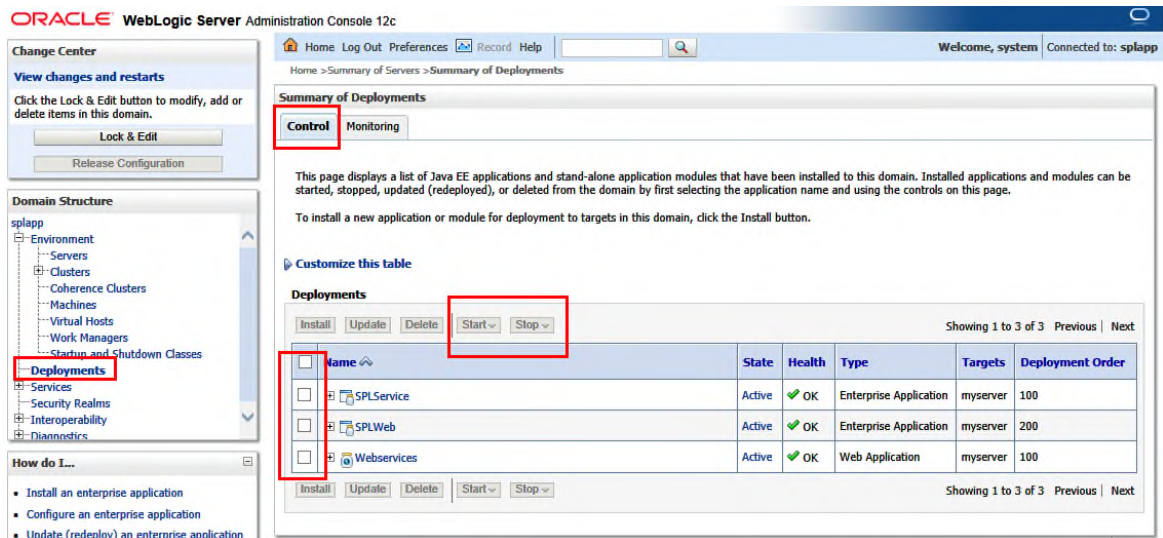


Figure 46 – Default Oracle WebLogic Deployments

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 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 |
| Webservices | 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.

²³ The default installation account is **system** if used.

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.

Specifying custom log file names **BATCH**

By default the [submitjob](#) and [threadpoolworker](#) utilities will create logs in a specific location. For example:

| User Exit | Platform | Default Location and Name |
|--------------------------------------|----------|---|
| submitjob.sh | Unix | <code>\$SPLOUTPUT/submitjob.{batchCode}.{sysDateTime}.log</code> |
| submitjob.cmd | Windows | <code>%SPLOUTPUT%\submitjob.{batchCode}.{sysDateTime}.log</code> |
| threadpoolworker.cmd | Windows | <code>%SPLOUTPUT%\threadpoolworker.{sysDateTime}.log</code> |
| threadpoolworker.sh | Unix | <code>\$SPLOUTPUT/threadpoolworker.{sysDateTime}.{pid}.log</code> |

Where:

| | |
|----------------------|--|
| {batchCode} | Batch Control used for job |
| {sysDateTime} | System Date and Time in YYYYMMDDHHmmSSSSS format |
| {pid} | Process Id |

If your implementation wishes to implement custom log file names then this may be achieved

using user exits which allow custom setting of the file name pattern. In the utilities an environment variable is set to the name and location of the log file. The user exit may be used to set this environment variable to an alternative. The user exit contains the script code fragment used to set the log file environment file name.

The table below lists the user exit, environment variable name and the platform:

| User Exit | Platform | User Exit Name | Env Variable |
|-----------------------------------|----------|---|-----------------------|
| <code>submitjob.sh</code> | Unix | <code>submitjob.sh.setvars.include</code> | <code>SBJLOGID</code> |
| <code>submitjob.cmd</code> | Windows | <code>submitjob.cmd.setvars.include</code> | <code>SBJLOGID</code> |
| <code>threadpoolworker.cmd</code> | Windows | <code>threadpoolworker.cmd.setvars.include</code> | <code>TPWLOGID</code> |
| <code>threadpoolworker.sh</code> | Unix | <code>threadpoolworker.sh.setvars.include</code> | <code>TPWLOGID</code> |

Additionally internal session variables are available for use in the user exits:

| Variable | submitjob ²⁴ | threadpoolworker | Comments |
|--------------------------|-------------------------|------------------|-------------------|
| <code>RUNOPTS</code> | | ■ | Runtime Options |
| <code>batchCode</code> | ■ | | Batch Code |
| <code>sysDateTime</code> | ■ | | Run Date and Time |

Note: Other environment variables in the session can be used and determined in the user exit script code.

Turning off L2 Cache **BATCH**

Note: This facility should only be used where background processes specifically require it. Turning off the cache in other circumstances will adversely affect performance.

By default, the threadpools use a batch cache to load common configuration data to avoid excessive calls to the database. In some cases it is desirable to disable the caching for a particular threadpool. To disable the cache on startup of the threadpool use the following command:

```
threadpoolworker[.sh] -l2 OFF -p <threadpoolname>
```

where `<threadpoolname>` is the name of the threadpool to start with the caching disabled.

Once the threadpool is started then all jobs that require caching off can be executed in this threadpool using the `DIST-THD-POOL` parameter (for online submission), `com.sp1wg.batch.submitter.distThreadPool` parameter in the job [specific properties file](#) or `-p` option on the `submitjob` command.

JDBC Support

Note: This facility is only available for online and web services channels only.

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

²⁴ These are set for the Java runtime.

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 using the *Services* → *Data Sources* menu item. Repeat for the connection pool for online and connection pool for web services. 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. It is recommended that the name be prefixed with jdbc/ to conform to naming standards. |
| Row Prefetch Enabled | This should be enabled . |
| Row Prefix Size | Specify a value between 2 - 100. Default: 48 |
| Stream Chunk Size | This is not used by the product |
| Database Type | Specify Oracle |
| Database Driver | Specify the database driver to use. <i>Do not use the XA versions of the driver for the product. By default use the Thin driver for Service Connections.</i> |
| 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 . If creating a separate connection pool for Web Services then it is recommended to use XAI_DBUSER in the ENVIRON.INI to separate traffic in the database. |
| Password | Specify the password for the user. This should correspond to DBPASS in the ENVIRON.INI . If creating a separate connection pool for Web Services then it is recommended to use XAI_DBPASS in the ENVIRON.INI to separate traffic in the database. |

| Attribute | Comment |
|----------------------|---|
| 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. |
| Statement Cache Type | This denotes the type of cache to use. Use the LRU setting. |
| Statement Cache Size | This is the cache size. |

- Specify the following values in the [ENVIRON.INI](#) using the [configureEnv](#) utility:

| Setting | Recommendation |
|--|--|
| IWS JDBC DataSource Name | Fully qualified JNDI Name for Web Services Data Source |
| Web JDBC DataSource Name | Fully qualified JNDI Name for Web Data Source |

- Execute the **initialSetup** utility to reflect the change.
- If using native mode, [redeploy](#) the product EAR files.

Cancelling Batch Processes Using JMX **BATCH**

While JMX can be used to obtain monitoring information it is possible to cancel threads of batch processes using the operations component of JMX. To cancel a thread the following process must be performed:

- Start the JMX console of your choice and connect to the relevant JMX port configured for the batch.
- Select the thread and batch process to be cancelled from the JMX console.
- Select the *Cancel* operation from the operations component of the console. The console may recognize the operations of the JMX classes and allow the actions to be processed. For example, **jconsole** will generate *cancelThread* button. Issue the action.

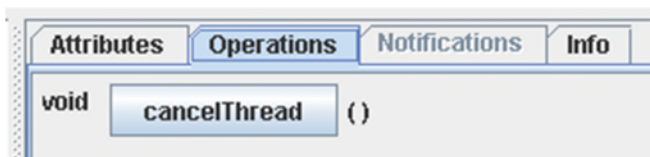


Figure 47 – Example cancelThread

Note: Depending on the JMX console used, a confirmation dialog may NOT be displayed and cannot be undone once issued. Ensure that the correct thread for the batch process is selected. To cancel a batch process, ALL threads must be cancelled.

The batch process will be marked as cancelled and stopped. The IP address of the requestor is logged in the Batch Run Tree for auditing purposes.

Operations from Oracle Enterprise Manager

Note: This section of the Server Administration Guide provides an overview of using Oracle Enterprise Manager along with the Application Management Pack for Oracle Utilities to administer the product.

Note: This section assumes that Oracle Enterprise Manager and the Application Management Pack for Oracle Utilities are installed prior to using this section. If this is not the case, refer to the installation documentation for these products to complete that prerequisite installation.

Discovering Oracle Utilities targets

Prior to using Oracle Enterprise Manager the Oracle Utilities targets must be discovered and registered with the Oracle Management Repository (OMR) within an Oracle Management Server (OMS) using Oracle Management Agents (OMA).

To discover the Oracle Utilities targets the following process must be followed:

- Install the Oracle Management Server and Oracle Management Repository to the relevant centralized server as per the Oracle Enterprise Manager documentation.

Note: It is not recommended to install the Oracle Management Server or Oracle Management Repository on the same servers housing the products.

- Deploy the Oracle Management Agents to the hosts housing the products.
- Discover and register the hosts and agents to the Oracle Management Repository using any of the methods described in the Oracle Enterprise Manager documentation.
- Start the products using the console or scripts.

Note: Oracle Enterprise Manager will only discover and register targets that are active at the time of discovery.

- Deploy the base Fusion Middleware functionality to the agents on the hosts for the products.
- Discover the Fusion Middleware targets associated with the product. This includes the domain, servers and application deployments.
- Download the Application Management Pack for Oracle Utilities via Oracle Software Delivery Cloud or Oracle Enterprise Manager Self Update.
- Deploy the Application Management Pack for Oracle Utilities to the Oracle Management Server to enable the server content.

Note: This may require a short outage on the Oracle Management Server.

- Deploy the the Application Management Pack for Oracle Utilities to the Oracle Management Agents on the product hosts to enable agent content.
- Discover the Oracle Utilities Environment target using the Discovery process as documented in the Oracle Enterprise Manager documentation. All related targets will be automatically discovered.

Note: At discovery time, target names are autogenerated. They can only be renamed with more meaningful names at discovery time.

After discovery of the Oracle Utilities Environment target the following Oracle Utilities targets are discovered, registered and associated with the Oracle Utilities Environment target:

- **Oracle Utilities Web Application** target - One or more Web Application Server targets (SPLWeb) associated with the Oracle Utilities Environment target. This target is associated with the Fusion Middleware Web Application Server base target.
- **Oracle Utilities Web Services** target - One or more Business Application Server targets (SPLService) associated with the Oracle Utilities Environment target. This target is associated with the Fusion Middleware Web Application Server base target.
- **Oracle Utilities Home** target - One or more software installations associated with the product associated with the Oracle Utilities Environment target.
- **Oracle Utilities Batch Server** target - One or more Coherence hosts associated with the Oracle Utilities Environment target.

Discovering Oracle WebLogic Targets

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:

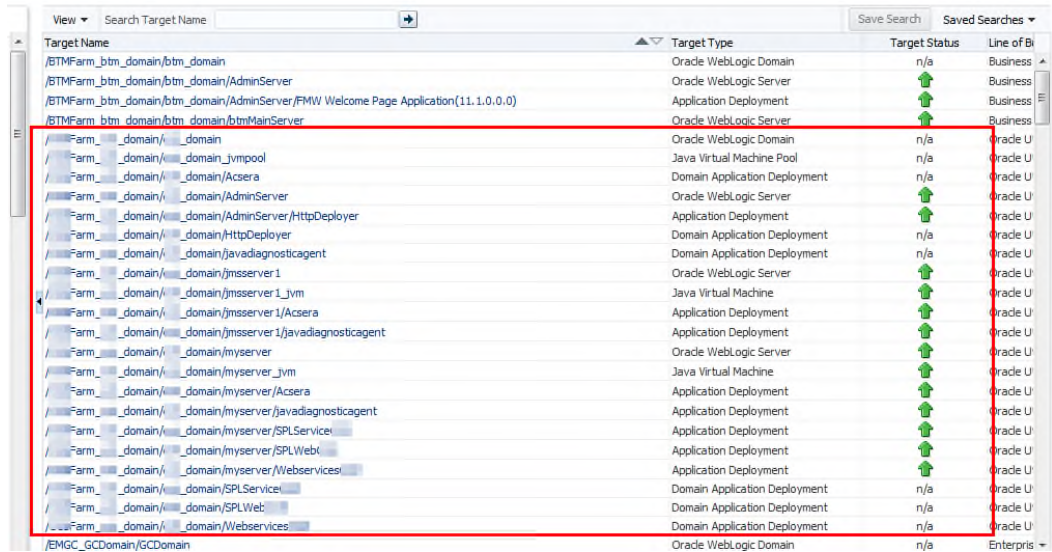


Figure 48 – Oracle WebLogic Targets

Oracle Utilities Targets

The following targets are available for the product:

| Target | Usage |
|---|---|
| Oracle Utilities System Environment | Environment Target as a collection of other targets |
| Oracle Utilities Home | Software installation(s) for environment. Used to cloning and patching |
| Oracle Utilities Web Application | Online Web Application(s) |
| Oracle Utilities Web Services Application | Business Application Server Application(s) |
| Oracle Utilities Batch Server | Threadpool/Submitter Batch Server(s) |

Refer to the [Oracle Application Management Pack for Oracle Utilities Overview](#) (Doc Id: [1474435.1](#)) for more information.

Monitoring using Oracle Enterprise Manager

The following target types within Oracle Enterprise Manager can be used with the product directly:

| Target | Monitoring capabilities |
|---|--|
| Oracle Utilities Web Application | Availability for online transaction including Oracle WebLogic basic metrics. |
| Oracle Utilities Web Services Application | Online performance monitoring |

| Target | Monitoring capabilities |
|-------------------------------|---|
| Oracle Utilities Batch Server | Batch performance monitoring |
| Oracle WebLogic Server | Availability, JMS, MDB and URL performance information |
| Application Deployment | Individual deployment metrics for Online, Business Application Server and Inbound Web Services including availability, performance and other metrics. |
| Java Virtual Machine | JVM Diagnostics (if Oracle WebLogic Management Pack EE installed) |
| Database Instance | Database metrics including performance and availability |

Note: It is not recommended to use the Oracle Coherence targets within Oracle Enterprise Manager for monitoring or managing the batch cluster.

Oracle Scheduler Integration

Note: The Oracle Scheduler interface is an optional installation for using the Oracle Scheduler in the Oracle Database to schedule and execute product jobs. Customers using alternative methods or tools can ignore this section.

Introduction

The product now includes optional interfaces to the Oracle Scheduler which is part of the Oracle Database. These interfaces allow the management of Oracle Scheduler objects and the execution of product batch processes using the scheduling capabilities of the Oracle Scheduler.

Background Materials

Before reading this section it is recommended to familiarize yourself with the Oracle Scheduler by reading the following documentation:

- [Oracle Scheduler Concepts](#) - Overview of the scheduler including the architecture.
- [Scheduling Jobs with Oracle Scheduler](#) - Creating, running and monitoring jobs, chains and prioritization.
- [Adminstrating Oracle Scheduler](#) - Configuration, monitoring, import/export schedules, troubleshooting etc
- [Using Oracle Scheduler with a CDB](#) - Using Oracle Scheduler in and/or with a multi-tenant container database.
- [DBMS_SCHEDULER Reference](#) - Reference API for the Oracle Scheduler including all the valid procedures available.

Note: The Oracle Scheduler in Oracle Database 12c and above is recommended to be used for this interface.

Oracle Scheduler Interface Overview

The product now includes REST services that lets users manage Oracle Scheduler objects using product services. This interface makes it easier to integrate batch schedules with enterprise schedulers.

- These are a set of services (invoked through REST APIs) to create, update, read, delete and execute Oracle Scheduler objects. These services are managed within the product itself. These services internally invoke necessary Oracle Scheduler functions to manage & monitor schedules and batch executions.
- Following set of OUAF business services can be invoked using REST to manage Oracle scheduler objects: (detailed interface definition is mentioned in [Oracle Scheduler – OUAF REST APIs sections](#))
 - F1-DBMSProgram - Add, read, change, or deletes a DBMS Scheduler Program object based on the action provided.

- F1-DBMSChain - Add, read, change, or deletes a DBMS Scheduler Chain object based on the action provided.
- F1-DBMSSchedule - Add, read, change, or deletes a DBMS Scheduler Schedule object based on the action provided
- F1-DBMSJob - Add, read, change, or delete a DBMS Scheduler Job object based on the action provided
- F1-DBMSList - Retrieves a list of names and descriptions of scheduler object based on the type provided and an optional name to filter with
- F1-DBMSSubmitJob - Submits a scheduler job for execution.
- F1-DBMSGetJobDetails - Retrieves the details of DBMS Scheduler Job based on job name
- F1-GetDBMSJobs - Retrieves a list of scheduler jobs based on status. The output is either a list of running jobs or a list of completed jobs based on the status provided.
- F1-DBMSCancelJob - Uses the service defined above to cancel a DBMS Scheduler Job.
- F1-DBMSRestartThread - Uses the service defined above to restart a batch job thread.
- F1-DBMSRestartChain - Uses the service defined above to restart DBMS Scheduler Chain.

In addition to the above, the product includes interfaces to allow Oracle Scheduler to schedule and execute product batch processes.

The components of interface are as follows:

- A set of tables to configure and control the interface. These tables are not managed by any object within the product itself as they are exclusively reserved for internal use by the Scheduler interface. These are installed in the product schema.
- A PL/SQL procedure ([OUAF BATCH](#)) that implements an interface to Oracle Scheduler to populate the scheduler interface tables, execute jobs and manage jobs from the PL/SQL interface supported by the Oracle Scheduler. This is installed on the product schema.
- The scheduler object may exist on the same database as the product or remotely on a scheduler database. If the remote option is used then the [Oracle Scheduler Agent](#)²⁵ must be installed and enabled on the product database servers. A [destination](#) and

²⁵ The Oracle Scheduler Agent is an option on the database installation or the Oracle Client installation.

[credentials](#) must also be configured.

- If [File Watchers](#) are used on product software hosts, then the [Oracle Scheduler Agent](#) must be enabled to access the files on the hosts holding the file.
- No user interface components are supplied with the interface from the product itself. The Oracle Scheduler uses Oracle SQL Developer or Oracle Enterprise Manager to maintain, execute and monitor Oracle Scheduler objects. It is recommended customers and partners use those interfaces to manage your schedule and related objects for Oracle Scheduler.
- For the Oracle Scheduler Interface to operate at least one threadpool.
- The Oracle Scheduler interface must be installed on the product schema but the schedule can be maintained locally or remotely. To support remote invocations, it is highly recommended to setup a dedicated database user for the schedule which is independent of the product database users. The product installation guide uses the default **SCHADM** for this role. All schedule information is stored in this schema not the product schema.

*Note: This manual will use the default value for this schedule administrator **SCHADM** but the actual implementation may vary.*

Oracle Scheduler Configuration

The Oracle Scheduler interface is installed on the product database and optionally on the scheduler database (for enterprise wide deployments).

After the Oracle Scheduler interface is installed the scheduler interface operates on the following principles:

- The Oracle database holding the schedule must have the **JOB_QUEUE_PROCESSES**²⁶ set to the maximum number of total concurrent threads²⁷ expected to execute at peak times to ensure maximum thread usage.
- The Oracle Scheduler and interface are configured to define the preferences to set the behavior:
 - The Oracle Scheduler preferences are configured with global scheduling defaults including email servers, time zones etc. For more information on the Oracle Scheduler preferences refer to [Setting Scheduler Preferences](#) and [Oracle Scheduler Preferences](#).
 - The Oracle Scheduler Interface options are set appropriately. For more information refer to [Scheduler Interface Options](#).
- The Oracle Scheduler is configured with the following minimal schedule objects. Additional objects can be configured as outlined in [Scheduling Jobs with Oracle Scheduler](#) and [Using Additional objects in Oracle Scheduler](#).
 - **Job** - Define one or more jobs per Batch Control using the [Submit Job](#) interface. Refer to [Defining Jobs](#) for additional interface advice.
 - **Chain** - Define the dependency chain with all the jobs that need to be executed

²⁶ The default is set to 1000 jobs, unless altered.

²⁷ This includes any DBA tasks and non-product processes that use the Oracle Scheduler at ANY time.

with that chain. Refer to [Defining Chains](#) for additional advice.

- Optionally, populate the parameters of the each job as outlined in [Specifying Job Parameters](#))

Oracle Scheduler Execution

Note: The threadpools used by any process must be started before any job can execute.

Note: Refer to the [Scheduler Architecture](#) of how the Oracle Scheduler internally operates.

Once the scheduler objects are defined the Oracle Scheduler will initiate the batch jobs using the following technique:

- When a job in the Scheduler is deemed ready to execute, the Oracle Scheduler Interface determines the parameters from the various objects (including from the PL/SQL command line) and creates entries in the batch instance tables.
- The product batch daemon detects a pending process and distributes the job to the threadpool(s) indicated on the job parameters for execution. The Oracle Scheduler does not distribute the jobs to the threadpools, that is done via the daemon.
- During execution the monitoring capabilities in Oracle Scheduler, Oracle Enterprise Manager, JMX or the Batch Run Tree can be used to monitor the process. Refer to [Monitoring Oracle Scheduler](#) for more information.
- If the batch process fails the interface set the job to the **FAILED** state for the Oracle Scheduler to react accordingly. If the batch process completes successfully the interface will set the job to the **SUCCEEDED** state for the Oracle Scheduler to react accordingly.

The following figure summarizes the flow and processing for the Oracle Scheduler, the Oracle Scheduler Interface and batch architecture:

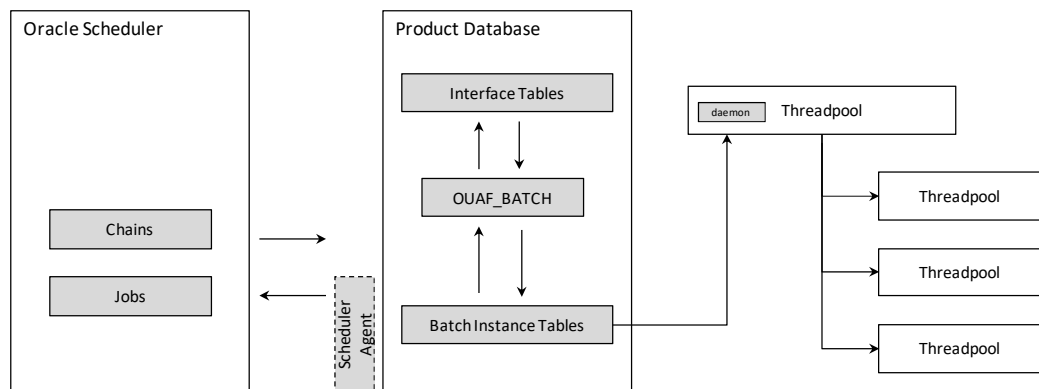


Figure 49 – Oracle Scheduler Interface

Oracle Scheduler Interface (OUAF_BATCH)

The Oracle Scheduler Interface is a PL/SQL based package **OUAF_BATCH** which contains a number of procedures to initiate batch processes, cancel running batch processes and manage the parameters for the interface.

Submit_Job - Submit a Job or Thread

The **Submit_Job** procedure submits a batch job and waits for it to complete or fail. It returns COMPLETED or FAILED state to the Oracle Scheduler when used in a Job or Program object.

| Column Name | Data Type | Req | Description |
|-------------------------|-----------|-----|---|
| batch_code | Text | ■ | Valid batch control |
| user_id | Text | ■ | Valid product user defined in user object. The email address specified on the user object is used to send any configured notifications. Value can be inherited from configured parameters. |
| thread_count | Number | | Override for thread count on batch control. |
| business_date | Date | | Process date to be used for batch job. Value can be inherited from configured parameters. Value can be set using the to_date function. If no value is provided, the current date will be used. The format of the date must be in a recognizable format. For example, DD-MMM-YYYY is supported. It is recommended to use the to_date function to avoid confusion. |
| max_timeout_mins | Number | | Maximum timeout, in minutes, for transactions to avoid ORA-01555 Snapshot too old errors . Default is 15 . |
| rerun_number | Number | | Used for rerunning specific extracts. Default is: 0 (zero) for current extract. |
| thread_pool | Text | | Valid Threadpool name to execute this batch job. This corresponds to the DIST-THD-POOL parameter. If not provided at all, the value of DEFAULT is assumed. |
| min_run_status | Number | | Minimum Run Status that will cause this procedure to exit. The default is RUN_STAT_COMPLETED (40). Other values supported: <ul style="list-style-type: none"> • 10 - RUN_STAT_PENDING • 20 - RUN_STAT_IN_PROGRESS |

| | | |
|-----------------------------|---------|---|
| | | <ul style="list-style-type: none"> • 30 - RUN_STAT_ERROR • 40 - RUN_STAT_COMPLETED |
| raise_error | Boolean | If the min_run_status is set a value less than 40 , this parameter is set to true and the job status matches the value of the min_run_status then raise an application error. This allows the chain to support conditional branches or stop chains on conditions. Default is false . |
| thread_notifications | Boolean | Whether if an error in any thread should trigger a notification event. Default is true . See Notification Options for more information. |
| soft_parm_map | Array | Associate array of soft parameters. Refer to Specifying Job Parameters for more details. |
| soft_parm_list | Array | Table array of soft parameters. Refer to Specifying Job Parameters for more details. |

Example command lines:

```

begin
  OUAF_BATCH.Submit_Job(
    batch_code => 'F1-SYNRQ',
    user_id => 'FRED01',
    thread_count => 4,
    business_date => '01-JUN-15',
    max_timeout_mins => 15,
    rerun_number => 0,
    thread_pool => 'DEFAULT',
    min_run_status => OUAF_BATCH.RUN_STAT_COMPLETED,
    raise_error => true,
    thread_notifications => true,
    soft_parm_list => OUAF_BATCH.parm_list_t('someParm','someValue'));
end;

begin
  OUAF_BATCH.Submit_Job(batch_code => 'F1-SYNRQ', user_id => 'FRED01');
end;

begin
  OUAF_BATCH.Submit_Job(batch_code => 'F1-SYNRQ',
    user_id => 'FRED01',
    business_date => to_date('2016-01-01','YYYY-MM-
DD'),
    thread_count => 1);
end;
begin

```

```

OUIAF_BATCH.Submit_Job(batch_code => 'F1-SYNRQ',
                      user_id => 'FRED01',
                      thread_pool => 'POOL01');

end;

begin
  OUIAF_BATCH.Submit_Job(
    batch_code => 'F1-SYNRQ',
    user_id => 'FRED01',
    soft_parm_list =>
      OUIAF_BATCH.parm_list_t('maintenanceObject', 'F1-SYNC REQ',
                              'isRestrictedByBatchCode', 'true'));

end;

begin
  OUIAF_BATCH.Submit_Job(batch_code => 'F1-SYNRQ',
                        user_id => 'FRED01',
                        business_date => to_date('2015-06-01', 'YYYY-MM-
DD'));
end;

```

Restart_Thread - Restart a failed thread/job

If a batch job or a thread of a job has failed and needs to be restarted the **OUIAF_BATCH** includes an operation to restart the thread.

The **Restart_Thread** procedure uses the same parameters of the **Submit_Job** procedure with an additional parameter:

| Column Name | Data Type | Req | Description |
|----------------------|-----------|-----|--|
| thread_number | Number | ■ | Thread Number to restart. For single threaded jobs, this should be set to 1 . |

The parameters can match the original execution or be altered to change the behavior of the individual thread.

*Note: The **thread_count** should match the original to ensure all data is processed correctly.*

The major difference between **Submit_Job** procedure and **Restart_Thread** procedure is that **Restart_Thread** procedure does not wait for the thread to complete. It simply ends after submitting the thread. Therefore it is not recommended to be used in a schedule in Oracle Scheduler.

Examples of the command are as follows:

```

begin
  OUIAF_BATCH.Restart_Thread(batch_code => 'F1-SYNRQ',
                            thread_number => 2,
                            user_id => 'FRED01',
                            thread_count => 4,
                            business_date => '01-JUN-15',
                            max_timeout_mins => 15,
                            rerun_number => 0,
                            thread_pool => 'POOL01');

end;

```

Cancel_Job - Cancel a running Job

Note: This technique is an alternative to the technique outlined in [Batch JMX Reference](#).

Note: The `dbms_scheduler.stop_job` procedure is recommended to stop a scheduler job.

A job that is executing can be stopped using the `Cancel_Job` procedure. The procedure can use one of the following identifiers to cancel the running job:

| Column Name | Data Type | Description |
|-------------------------|-----------|--|
| <code>batch_code</code> | Text | Valid batch control |
| <code>job_id</code> | Number | Allocated job identifier from batch daemon (on <code>ci_batch_job</code> table). |

For example:

```
begin
  OUAF_BATCH.Cancel_Job(job_id => '0000000000123',
                        batch_code => 'F1-SYNRQ');
end;
```

```
begin
  OUAF_BATCH.Cancel_Job(job_id => '0000000000123');
end;
```

```
begin
  OUAF_BATCH.Cancel_Job(batch_code => 'F1-SYNRQ');
end;
```

The `Cancel_Job` procedure asynchronously changes the running `Submit_Job` status to `CANCELLED` state and terminates. The `Submit_Job` instance continually monitors the status and will take appropriate action to cancel all the running threads and end with a cancellation error once all the threads have ended.

Refer to [Cancellation](#) for more details.

Set_Option - Set a global or Job option

The `Set_option` procedure is used to set global or job specific options for the scheduler interface. Any option specified using this procedure is subject to the [Order Of Preference](#) for parameters.

The `Set_option` procedure uses the following command format:

```
begin
  OUAF_BATCH.Set_option(scope => '<scope>', name => '<parmname>', value
=> '<parmvalue>');
end;
```

<scope> Scope of the parameter. The valid values are **GLOBAL** for global settings or a valid Batch Control for a specific Batch Control. Settings at the Batch Control level override the **GLOBAL** scope.

<parmname> The parameter name from the list below. This setting can be overridden on the command line.

<parmvalue> The value of the parameter designated by **<parmname>** to be used for the

scope designated by *<scope>*.

The parameters supported by this option are listed below:

| Parameter Name | Description |
|-----------------------------|--|
| business_date | The default business date is the database's current system date, but this can provide an alternative default. It is typically useful in the case of jobs that run past midnight to make it maintain the same date for all the submissions. The date format is ISO format: YYYY-MM-DD . |
| user_id | The user id is required for every submission but to avoid having to specify it on every Submit_Job call, this will define a default either globally or for specific batch codes. The user must be defined as an user object within the product. |
| poll_seconds | By default the batch tables are checked every 1 second for updates. This provides for very responsive job handling, but if it is too taxing on the database, the frequency can be modified with this setting. The value must be between 1 and 60. |
| notify_job_name | Oracle Scheduler job OUIF_NOTIFY is the default thread notification job. That is defined out of the box to send emails for every failed thread. If that behavior is undesired, a custom thread notification, Oracle Scheduler job can be created and it defined as the default handler with this option. The new notify Oracle Scheduler job must exist. |
| thread_notifications | Set to false to suppress thread notifications for failed threads. The default is true . |
| thread_pool | The default threadpool to use. The installation default threadpool is 'DEFAULT' . |
| discard_queue | Set to true to delete all unused online job queue entries (OUI_BATCH_JOB rows) created by OUI_BATCH . The default is false . OUI_BATCH uses individual OUI_BATCH_JOB rows for the threads so it may become quite voluminous, and the job queue entries are typically not useful once a thread has started executing as the <i>Batch Run Tree</i> entries are used to track the job. This option can be used to clean up the unused entries. |
| debug | Set to true to see internally generated DBMS_OUTPUT debug messages. The default is false . Used for development purposes. |

For example:

```

Begin
  OUIF_BATCH.Set_Option(scope => 'GLOBAL', name => 'user_id', value =>
    'FRED01');

```

end

To remove the option, use the [Unset_Option](#) procedure.

Unset_Option - Reset a global or Job option

By using the [Set_Option](#) feature sets the global or job specific parameters for the scheduler it is possible to also remove the settings using the [Unset_Option](#) feature on the parameters covered by the [Set_Option](#) feature. The format of the command is:

```
begin
  OUAF_BATCH.Unset_Option(scope => '<scope>', name => '<parmname>');
end;
```

<scope> Scope of the parameter to reset. The valid values are **GLOBAL** for global settings or a valid Batch Control for a specific Batch Control.

<parmname> The parameter name to reset. Refer to the [Set_Option](#) procedure for the valid list.

For example:

```
begin
  OUAF_BATCH.Unset_Option(scope => 'GLOBAL', name => 'business_date');
end;
```

Raise_Thread_Error - Raise an error

Whenever a thread or job fails the Oracle Scheduler Interface informs the product of the failure as well as the scheduler itself. This is handled by the [Raise_Thread_Error](#). This procedure is not directly called by the scheduler itself and is used internally by the interface itself.

Setting Preferences

One of the first steps in configuration is setting the preferences

- The Oracle Scheduler includes a number of key preferences that need to be configured to ensure a consistent operation of the scheduler. These preferences are documented in the [Setting Scheduler Preferences](#) section of the [Oracle Scheduler Administration](#) documentation. It is recommended to set the preferences before using any of the functionality in the Oracle Scheduler.
- Once the Oracle Scheduler Interface is installed the **GLOBAL** options for the interface, should be set if they differ from the recommended defaults. This can be achieved using the [Set_Option](#) procedure. At a minimum, it is recommended to set a global **user_id** to act as a default to avoid individual job issues.
- Optionally, it is possible to set Batch Job specific options using the Batch Control as the key for the [Set_Option](#) procedure.

The above facilities set the overall scheduler and interface preferences.

Order Of Preference

There are a number of different techniques available for setting the parameters for any job. The available techniques from lowest precedence to highest are as follows:

- **Batch Control** - Job parameters can be set on the Batch Control record within the user interface under control of authorized users. This is commonly maintained by the business.
- **Global Preferences** - Some of the parameters from the Batch Control can be specified as a **GLOBAL** preference using the [Set Option](#) procedure. These preferences, if set, will override any Batch Control settings for the same setting.
- **Batch Job Preferences** - The [Set Option](#) procedure also supports the setting of parameters using the Batch Control as the key. Any overlapping settings set at this level will override settings on the Batch Control and any **GLOBAL** preference for the same setting.
- **Command Line Options** - It is possible to override all settings using command line options on the [Submit Job](#) and [Restart Thread](#) procedures. This is handy for special processing. See [soft_parm_map](#) and [soft_parm_list](#) for advice and examples.

It is recommended to set the values in the relevant level and use command line options for special situations or when using dynamic parameters.

Using soft_parm_map

Batch parameters can be specified on the command line through the [soft_parm_map](#), which is an [associative array](#). This format can be used in isolation or in association with [soft_parm_list](#) when specifying command line parameters. When both are used, the [soft_parm_list](#) values are taken in precedence.

To use [associative arrays](#), the array must be declared before use in the PL/SQL command.

Any array must be declared as type `OUIF_BATCH.parm_map_t`.

The format of soft parameter in an [associate array](#) is:

| Segment | Format |
|----------------------|--|
| Declaration of array | <code>declare <arrayname> OUIF_BATCH.parm_map_t;</code> |
| Definition of value | <code><arrayname>('<parmname>') := '<parmvalue>';</code> |
| Use of map | <code>soft_parm_map => <arrayname></code> |

where:

| | |
|--------------------------|--|
| <arrayname> | Name of associative array to be used as the <code>soft_parm_map</code> |
| <parmname> | Name of the parameter to be set |
| <parmvalue> | Value of parameter |

For example:

```

declare
  my_soft_parms OUAF_BATCH.parm_map_t;
begin
  my_soft_parms('maintenanceObject') := 'F1-SYNC REQ';
  my_soft_parms('isRestrictedByBatchCode') := 'true';
  OUAF_BATCH.Submit_Job(batch_code => 'F1-SYNRQ',
                        user_id => 'FRED01',
                        soft_parm_map => my_soft_parms);
end;

```

The **my_soft_parms** is the **soft_parm_map** in the above example. The use of associative arrays is a way of supporting dynamic parameters for complex parameters. This technique can be used exclusively or in conjunction with [soft_parm_list](#).

Note: The associative array should not contain common parameters such as business_date, user_id etc

Using soft_parm_list

An alternative to the [soft_parm_map](#) is to explicitly specify the parameters as a list of parameter and value pairs. This method is recommended for explicit values for the parameters. The parameter list must be of type **OUAF_BATCH.parm_list_t**.

The parameters are specified in the following format:

```
OUAF_BATCH.parm_list_t('<parmname>', '<parmvalue>'...)
```

where:

| | |
|--------------------------|---------------------------------|
| <parmname> | Name of the parameter to be set |
| <parmvalue> | Value of parameter |

For example:

```

begin
  OUAF_BATCH.Submit_Job(
    batch_code => 'F1-SYNRQ',
    user_id => 'SYSUSER',
    soft_parm_list =>
      OUAF_BATCH.parm_list_t('maintenanceObject', 'F1-SYNC REQ',
                             'isRestrictedByBatchCode', 'true'));
end;

```

This technique can be used exclusively or in conjunction with [soft_parm_map](#).

Note: The array should not contain common parameters such as business_date, user_id etc

Oracle Scheduler – OUAF REST APIs

The below are the list of OUAF services that can be used for managing Oracle DBMS scheduler objects:

1. F1-DBMSPROGRAM service:

This service can be used to Add, read, change, and delete a DBMS Scheduler Program object. There are currently two types of program that can be created:

- 1) Program associated with a batch code
- 2) Program to set/unset options

If a list of options is provided, it will create a program to set/unset options. Otherwise, it will create a program associated with a batch code

| Parameter Name | Description |
|-----------------------|---|
| Action | Specifies the action to be performed on the scheduler object. Valid values are: ADD – Add READ – Read CHNG – Change DEL – Delete |
| Name | The name of the scheduler program on which the operation needs to be performed. |
| description | Description of the scheduler program. |
| batchCode | Batch code of the OUAF batch process for which the program is being created. |
| userId | ID under which the batch process will run. |
| threadCount | Use Thread Count to control whether a background process is run single threaded or in multiple parallel threads. This value defines the total number of threads that have been scheduled |
| businessDate | Process date to be used for batch job. If no value is provided, the current date will be used. The format of the date must be in a recognizable format. |
| maxTimeoutMins | Maximum timeout, in minutes, for transactions to avoid ORA-01555 Snapshot too old errors. Default is 15. |
| rerunNumber | Is only used for OUAF background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run). |
| threadPool | The default threadpool to use. The installation default threadpool is DEFAULT . |
| minRunStatus | Minimum Run Status that will cause this procedure to exit. The default is RUN_STAT_COMPLETED (40). Other values supported: <ul style="list-style-type: none"> • 10 - RUN_STAT_PENDING |

| Parameter Name | Description |
|--|--|
| | <ul style="list-style-type: none"> 20 - RUN_STAT_IN_PROGRESS 30 - RUN_STAT_ERROR 40 - RUN_STAT_COMPLETED |
| raiseError | If the min_run_status is set a value less than 40, this parameter is set to true and the job status matches the value of the min_run_status then raise an application error. This allows the chain to support conditional branches or stop chains on conditions. Default is false. |
| threadNotifications | Whether if an error in any thread should trigger a notification event. Default is true. See Notification Options for more information. |
| parmName (list parameter) | The name of the parameter to the OUAF batch process, if applicable. |
| parmValue (list parameter) | The corresponding value of the parameter to the OUAF batch process, if applicable. parmValue and parmName pairs should be defined in repeated lists. |
| optionName (list parameter) | Option name to set/unset before program is executed |
| optionScope (list parameter) | Corresponding scope for the options to set/unset before program is executed. The scope can be global or per batch code. |
| optionValue (list parameter) | Corresponding value for the option name. |

2. F1-DBMSChain service:

Add, read, change, or deletes a DBMS Scheduler Chain object based on the action provided.

| Parameter Name | Description |
|-------------------------------------|---|
| action | Specifies the action to be performed on the scheduler object. Valid values are: ADD – Add READ – Read CHNG – Change DEL – Delete |
| name | The name of the scheduler chain on which the operation needs to be performed. |
| description | Description of the scheduler chain. |
| stepName (list parameter) | The name of the step within the scheduler chain |

| Parameter Name | Description |
|--|---|
| programName (list parameter) | The program name that is part of the corresponding step. |
| condition (list parameter) | A condition that must be met to complete the step and to transition to the next step or to end the chain execution. |

Note: A step called 'END' step must be included in the chain definition that includes the condition to meet for the chain to end.

3. F1- DBMSSCHEDULE service:

Add, read, change, or deletes a DBMS Scheduler's schedule object based on the action provided.

| Parameter Name | Description |
|-----------------------|---|
| action | Specifies the action to be performed on the scheduler object. Valid values are: ADD – Add READ – Read CHNG – Change DEL – Delete |
| name | The name of the scheduler's schedule object being created/edited/read/deleted. |
| description | Description of the schedule. |
| repeatInterval | The schedule's repeat interval to be specified in the format supported by Oracle DBMS Scheduler. (Refer to Oracle DBMS scheduler's documentation) |

4. F1-DBMSJOB service:

Add, read, change, or deletes a DBMS Scheduler's job object based on the action provided.

- The job can be associated with either a program or a chain by populating the appropriate field. A validation error will result if both or neither of the fields are populated.
- If schedule name is not populated, it will be for immediate execution.
- When called for change action, the job is added if it does not exist, else it is replaced with new definition.

| Parameter Name | Description |
|---------------------|---|
| action | Specifies the action to be performed on the scheduler object. Valid values are: ADD – Add READ – Read CHNG – Change DEL – Delete |
| name | The name of the scheduler's job object being created/edited/read/deleted. |
| description | Description of the scheduler's job object. |
| programName | The name of the scheduler's program to which this job should be associated with. |
| chainName | The name of the scheduler's chain to which this job should be associated with. |
| scheduleName | The name to the schedule's schedule object based on which this job needs to be executed. |
| Enabled | Specified if this job should be enabled or disabled. |

5. F1-DBMSJOB service:

This service returns a list of name and descriptions of DBMS Scheduler objects based on the type and name (if specified) provided.

| Parameter Name | Description |
|----------------|---|
| type | Specifies the type of scheduler object whose list is to be retrieved. Valid values are: PROG – DBMS Scheduler Program CHAN – DBMS Scheduler Chain SCHD – DBMS Scheduler Schedule JOB – DBMS Scheduler Job |
| name | Name to search with. Appending '%' to the end of name will perform a 'like' query resulting in a list of objects starting with the name provided. |

6. F1-DBMSSUBMITJOB service:

This service submits a scheduler job for execution.

| Parameter Name | Description |
|----------------|---|
| name | Name of the scheduler's job object to be submitted for execution. |

7. F1-DBMSGETJOBDETAILS service:

Retrieves the details of DBMS Scheduler job based on job name. More specifically, this retrieves the steps and the step state of the DBMS scheduler job.

| Parameter Name | Description |
|---|---|
| name | Name of the scheduler's job object whose details are to retrieved/read. |
| status | Status of the DBMS scheduler job. |
| startDate | The start date of the DBMS scheduler job. |
| runDuration | The run duration of the DBMS scheduler job. |
| stepName (list parameter of "steps" list) | The name of the DBMS Scheduler job's step. |
| status (list parameter of "steps" list) | The status of the corresponding step. |
| runDuration | The run duration of the corresponding step. |

8. F1-DBMSGETJOBS service:

This service retrieves a list of scheduler jobs based on status of running or completed. A date range can be specified to limit the list.

Note: The maximum numbers of records returned for completed jobs is 100.

This retrieves list of jobs for COMPLETED jobs, list of jobs with steps for RUNNING chains

| Parameter Name | Description |
|---|---|
| jobStatus | The status based on which the list of DBMS scheduler jobs list has to be retrieved. Valid values are: RUN – Running Jobs COMP – Completed Jobs |
| startDate | The start date of the date range for which the details have to be retrieved. |
| endDate | The end date of the date range for which the details have to be retrieved. |
| jobStatus (list parameter of “jobs” list) | The status of the job. |
| runDate (list parameter of “jobs” list) | The execution date of the corresponding job. |
| runDuration (list parameter of “jobs” list) | The run duration of the corresponding job. |
| chainName (list parameter of “jobs” list) | The name of the associated chain (if applicable) |
| stepName (list parameter of “steps” list) | The name of the chain step within the job. |
| stepState (list parameter of “steps” list) | The state of the corresponding step. |

9. F1-DBMSCANCELJOB service:

This service cancels a DBMS schedule job if it is running.

| Parameter Name | Description |
|----------------|---|
| name | The name of the DBMS scheduler job to cancel. |

10. F1-DBMSRESTARTTHREAD service:

This service can be used for restarting a failed OUAF batch thread.

| Parameter Name | Description |
|---------------------------------------|---|
| batchCode | Batch code of the OUAF batch process for which the failed thread needs to be restarted. |
| userId | User Id of the user using which the batch thread needs to be restarted. |
| threadNumber | The thread number of the batch program that needs to be restarted. |
| threadCount | Use Thread Count to control whether a background process is run single threaded or in multiple parallel threads. This value defines the total number of threads that have been scheduled |
| businessDate | Process date to be used for batch job. If no value is provided, the current date will be used. The format of the date must be in a recognizable format. |
| maxTimeoutMins | Maximum timeout, in minutes, for transactions to avoid ORA-01555 Snapshot too old errors. Default is 15. |
| rerunNumber | Is only used for OUAF background processes that download information that belongs to given run number. It should only be supplied if you need to download an historical run (rather than the latest run). |
| threadPool | The default threadpool to use. The installation default threadpool is 'DEFAULT'. |
| parmName (list parameter) | The name of the parameter to the OUAF batch process, if applicable. |
| parmValue (list parameter) | The corresponding value of the parameter to the OUAF batch process, if applicable. parmValue and parmName pairs should be defined in repeated lists. |

11. F1-DBMSRESTARTCHAIN service:

This service can be used to restart DBMS scheduler Chain. The service determines the failed step and changes status so the chain can resume

| Parameter Name | Description |
|----------------|---|
| jobName | The name of the DBMS scheduler job to cancel. |

Building a Schedule

Note: This section summarizes the features of the scheduling jobs in Oracle Scheduler using the Oracle Scheduler integration. Refer to [Scheduling Jobs with Oracle Scheduler](#) for additional information.

To use the Oracle Scheduler the scheduler objects must be loaded and the command lines use the provided Oracle Scheduler Interface PL/SQL package.

Recommended tools

There are a number of tools that are recommended to be used to define, maintain and manage the Oracle Scheduler:

- **PL/SQL interface** - There is a PL/SQL based command line interface that can be used to define the scheduler objects and control the scheduler via the **DBMS_SCHEDULER** package from any SQL interface tool. For more information about the PL/SQL interface, refer to the [DBMS_SCHEDULER Reference](#). This is useful for direct access environments. It is also used by the other tools, via generated statements, to manage the Oracle Scheduler.
- **Oracle SQL Developer** - Oracle SQL Developer includes an interface to manage scheduler objects including wizards to define key objects. For example:

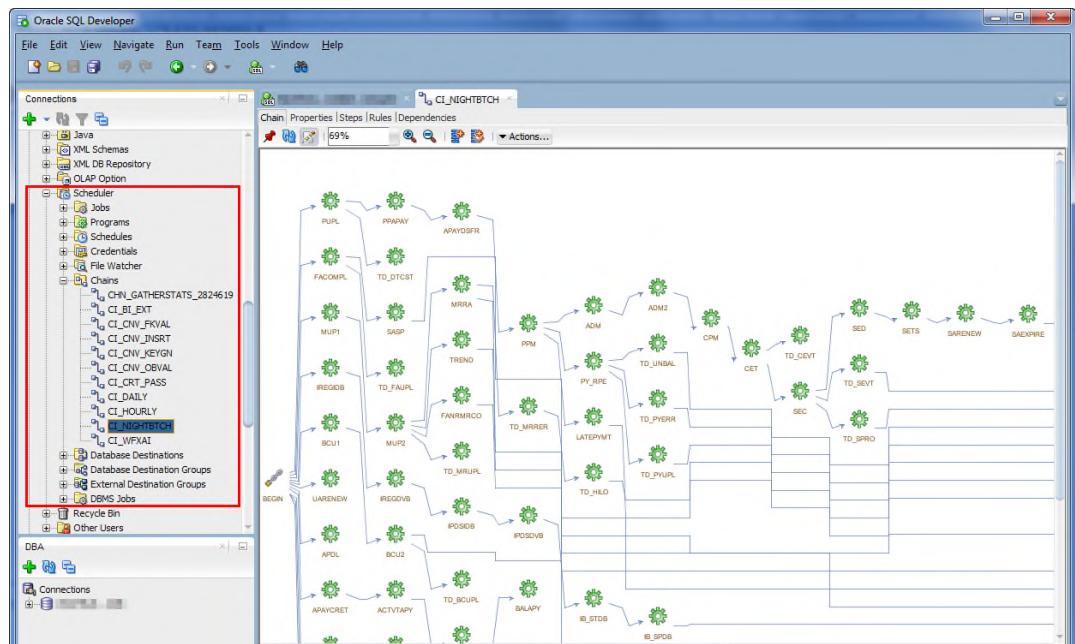


Figure 50 – Example Oracle SQL Developer Oracle Scheduler interface

- **Oracle Enterprise Manager** - The database targets associated with the Oracle Scheduler allow for the definition, management and monitoring scheduler objects. For example:

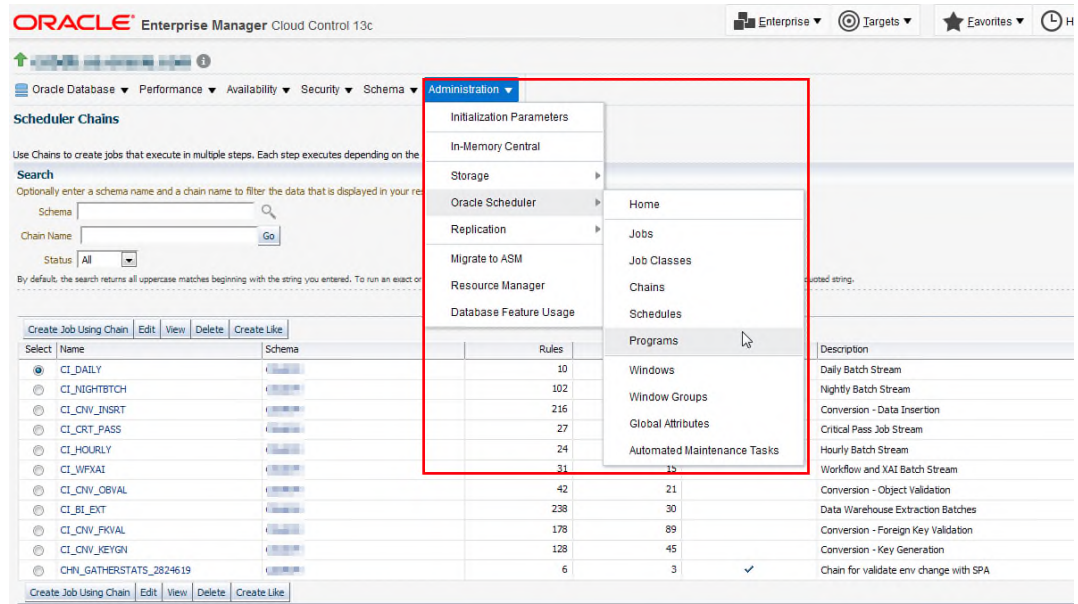


Figure 51 – Example Oracle Enterprise Manager Oracle Scheduler interface

It is possible to use any of the interfaces above to manage your schedule objects. Refer to the help pages associated with each tool for additional advice on individual objects.

Defining Programs

Note: Definition of Program Objects is optional and can be replaced directly with Job definitions. It is recommended that Program Objects be used for maximum reuse.

The first step in the process is to define each batch control as a Program object in the scheduler using the following attributes:

| Attribute | Recommended content |
|-----------------------|---|
| program_name | Name of the Program. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. |
| comments | A short description of the program. This is useful for searching the program. |
| enabled | Set to true to enable this program to be used by the scheduler. |
| program_type | Specify the PL/SQL Block as the type of program for the Oracle Scheduler Interface. |
| program_action | Specify the interface commands line for the execution. You may include any overrides, if desired, at this time. For optimization purpose, only specify the parameters that are not inherited from the global options or are not defaulted on the batch control. It is recommended to specify the Batch Control to avoid issues in submission. An example of a simple command line is: begin OUF_BATCH.Submit_Job(batch_code => |

| Attribute | Recommended content |
|-----------|---------------------|
| | 'BATCH01'); end; |

There are number of ways that can be used to specify the program using the various tools available:

| Interface | Method of Definition |
|--------------------|---|
| PL/SQL | Use the DBMS_SCHEDULER.CREATE_PROGRAM procedure using the above parameters to define the program object. |
| SQL Developer | Use the <i>New Program</i> menu item of the <i>Scheduler</i> tree to define the new program using a right click action on the <i>Program</i> node. |
| Enterprise Manager | Using the <i>Administration</i> → <i>Oracle Scheduler</i> → <i>Programs</i> menu item, specify the program information as above. Use the Source field to hold the PL/SQL command line for the interface. |

You may create more than one program for the same batch control if there are different parameters is needed for different executions.

Defining Job Classes

*Note: The use of [Job Classes](#) is optional. It is only used to group jobs for attribute inheritance, service affinity, monitoring purposes and/or resource profiling. **DEFAULT_JOB_CLASS** is used if no job class is specified.*

Note: Job Class is only one technique for prioritizing jobs. Refer to [Prioritizing Jobs](#) for more information.

If [Job Classes](#) are to be used, they need to be setup to be used by the jobs allocated to them. This attributes of a Job Class are as follows:

| Attribute | Recommended content |
|-------------------------------|---|
| job_class_name | Name of the Job Class. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. |
| comments | A short description of the job class. This is useful for searching for the job class. |
| resource_consumer_name | Name of the resource consumer group associated with jobs in this job class. Optional. |
| service | Name of database service associated with this job class name. Optional. |
| logging_level | Sets the level of logging is performed by the Oracle Scheduler. Optional. |
| log_history | Sets the retention period, in days, log history are retained by the Oracle Scheduler. Optional. |

There are number of ways that can be used to specify the job class using the various tools available:

| Interface | Method of Definition |
|--------------------|--|
| PL/SQL | Use the DBMS_SCHEDULER.CREATE_JOB_CLASS procedure using the above parameters to define the job class object. |
| SQL Developer | Use the <i>New Job Class</i> menu item of the <i>Scheduler</i> tree to define the new job class using a right click action on the <i>Scheduler</i> node. |
| Enterprise Manager | Using the <i>Administration</i> → <i>Oracle Scheduler</i> → <i>Job Class</i> menu item, specify the class information as above. |

Defining Schedules

Note: This section is only a subset of the capabilities of the schedule object. For more advanced discussions and different scenarios refer to the [Creating and Managing Schedules to Define Jobs](#) in the Oracle Database documentation.

Before defining jobs and chains you may need to define a schedule to run them against. Schedules can be time based and/or event based. It is also possible to create multiple schedules to support complex scheduling requirements. For example, you can create a holiday schedule with the dates that are blacklisted for execution in another schedule.

The following attributes are typically populated for the schedule object:

| Attribute | Recommended content |
|------------------------|--|
| schedule_name | Name of the schedule. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. |
| comments | A short description of the schedule. This is useful for searching the schedule. |
| start_date | Optional start date and time when the schedule becomes valid. If not populated then the date jobs or windows are activated is used. |
| repeat_interval | This specifies how often the schedule repeats. It uses the Calendaring Syntax supported by the Oracle Scheduler. Even schedules decide the repeat interval for checking the event. |
| end_date | Optional date and time after jobs and windows allocated to this schedule will not execute. |

There are number of ways that can be used to specify the schedule using the various tools available:

| Interface | Method of Definition |
|---------------|--|
| PL/SQL | Use the DBMS_SCHEDULER.CREATE_SCHEDULE procedure using the above parameters to define the schedule object. This interface is recommended for complex Calendaring Syntax support. |
| SQL Developer | Use the <i>New Credential</i> menu item of the <i>Scheduler</i> tree to define |

| Interface | Method of Definition |
|--------------------|--|
| | the new schedule using a right click action on the <i>Schedule</i> node. |
| Enterprise Manager | Using the <i>Administration</i> → <i>Oracle Scheduler</i> → <i>Schedules</i> menu item, specify the schedule information as above. |

Defining Remote Destinations and Credentials

*Note: This section outlines the process of specifying destinations and credentials for remote processing. If the database scheduler is housed within the product database (the default installation) then this section only applies to **File Watcher** type jobs.*

*Note: In Oracle Database 12c, the **CREATE_CREDENTIAL** procedure for **DBMS_SCHEDULER** has been deprecated and replaced with the **CREATE_CREDENTIAL** procedure within the **DBMS_CREDENTIAL** package.*

When executing Jobs and File Watchers that are remote from the scheduler database and location, it is required that a destination be defined for the job and relevant credentials for the job to execute on that destination. The following needs to be setup:

- To execute product jobs remotely, the product database server needs to be defined as a destination and the product database credentials should be specified. It is recommended that the values expressed in **BATCH_DBUSER** and **BATCH_DBPASS** be specified as credentials.
- For **File Watcher** jobs, the product server where the file is housed must be defined as a destination. The operating system credentials for the batch process must be specified to access the file must be specified as credentials.
- There is no explicit process for creating non-database related destinations in the API (there is no **CREATE_AGENT_DESTINATION** procedure). When an agent registers with the database scheduler as part of the startup procedure, it is automatically added as a destination. If you wish to register remote databases then use the **CREATE_DATABASE_DESTINATION** package.
- For [Database Destinations](#), the database TNS information must exist on the scheduler database to connect remotely.
- At least one product related database destination must be registered for remote invocation of product jobs.
- The [Oracle Scheduler agents](#) must be executing for the scheduler to initiate and track remote jobs.

To specify the credentials the following [attributes](#) must be provided:

| Attribute | Recommended content |
|------------------------|--|
| credential_name | Name of the credential. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. |
| comments | This attribute is not used by the DBMS_SCHEDULER package but can be provided for documentation purposes. |
| username | The userid to be used for connection. This attribute is required |

| Attribute | Recommended content |
|-----------------------|---|
| | and is case sensitive. |
| password | The password associated with the username. This value is stored in obfuscated format and is not available in Oracle Scheduler views. |
| database_role | This is the database role associated with this user. This is not used for product jobs but is required for any DBA tasks requiring privileged access. |
| windows_domain | For Windows Server implementations only. This is the windows domain for the user to connect to. |

There are number of ways that can be used to specify the credentials using the various tools available:

| Interface | Method of Definition |
|--------------------|--|
| PL/SQL | Use the DBMS_CREDENTIAL.CREATE_CREDENTIAL procedure using the above parameters to define the credential object. |
| SQL Developer | Use the <i>New Credential</i> menu item of the <i>Scheduler</i> tree to define the new credential using a right click action on the <i>Credentials</i> node. Oracle SQL Developer uses DBMS_CREDENTIAL for credential management. |
| Enterprise Manager | Oracle Enterprise Manager uses its own security system for credentials. Refer to the Enterprise Manager Cloud Control Security Guide for more details. |

When specifying [Database Destinations](#), the following attributes need to be specified:

| Attribute | Recommended content |
|-------------------------|---|
| destination_name | Name of the destination. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. |
| comments | This attribute is not used by the DBMS_SCHEDULER package but can be provided for documentation purposes. |
| agent | The agent name as specified in the agent configuration file. |
| tns_name | Oracle service name as documented in the TNS file attached to the configuration file. This is used to connect from the scheduler database to the remote database. |

There are number of ways that can be used to specify the database destination using the various tools available:

| Interface | Method of Definition |
|-----------|---|
| PL/SQL | Use the DBMS_CREDENTIAL.CREATE_DATABASE_DESTINATION procedure using the above parameters to define the credential |

| Interface | Method of Definition |
|--------------------|---|
| | object. |
| SQL Developer | Use the <i>New Database Destination</i> menu item of the <i>Scheduler</i> tree to define the new destination using a right click action on the <i>Database Destinations</i> node. |
| Enterprise Manager | Oracle Enterprise Manager uses its own security system for destinations. Refer to the Enterprise Manager Cloud Control Security Guide for more details. |

Defining Jobs

Note: This section is only a subset of the capabilities of the job object. For more advanced discussions refer to the [Jobs](#) section in the Oracle Database documentation.

Note: Product jobs should never be defined as Lightweight jobs. Those are reserved for specific Oracle products.

The key object in the Oracle Scheduler is the [Job](#) object. This object is a collection of data that tells the scheduler what, when, where and how to execute a particular process. In the case of the product, a job defines how to execute an instance of a Batch Control whether it is independent or part of a Chain.

Each Job definition has a set of attributes that define the Job:

| Attribute | Recommended content |
|----------------------------|--|
| job_name | Name of the job. This name should not include any embedded characters or '-' character according to Oracle object naming conventions. It is recommended to replace '-' in Batch Controls with '_' to avoid naming issues. |
| comments | Description of Job. Useful for finding jobs. |
| job_type | If programs have been used then the type STORED_PROCEDURE should be used. If programs are not used, then specify PLSQL_BLOCK with the action containing the source PL/SQL command. |
| job_action | Value depends on the specification of the job_type : <ul style="list-style-type: none"> If job_type is STORED_PROCEDURE, then this will contain the fully qualified name of the program to execute. If job_type is PLSQL_BLOCK, then this will contain the PL/SQL command to execute the OAAF_BATCH interface. |
| start_date | Optional start date and time when the schedule becomes valid. If not populated then the date jobs or windows are activated is used. If schedule_name is specified, then this date and time may be null to inherit the value from the schedule or chain. |

| Attribute | Recommended content |
|-------------------------|---|
| repeat_interval | Optional. Definition of how often the job will repeat. It uses the Calendaring Syntax supported by the Oracle Scheduler. Even schedules decide the repeat interval for checking the event. If schedule_name is specified, then this value may be null to inherit the value from the schedule or chain. |
| schedule_name | Optional. Name of schedule object dictating the schedule for this job. |
| job_class | Optional. Name of job class allocated to this job. |
| end_date | Optional date and time after jobs and windows allocated to this job will not execute. If schedule_name is specified, then this date and time may be null to inherit the value from the schedule or chain. |
| credential_name | Optional. Name of credential to use for remote invocation for this job. |
| destination_name | Optional. Name of destination to use for remote invocation for this job. For remote product jobs this must be the location of the product database. |
| enabled | Whether this job is enabled or not. Only enabled jobs (true) are executed. |
| auto_drop | Whether after execution of this job, the job definition will be dropped. This is typically set to true for temporary jobs only. It is recommended to set to false for all regular product jobs. |

There are additional attributes to manage the advanced features of the job. Refer to the [Job Object](#) for a discussion of the additional parameters.

There are number of ways that can be used to specify jobs using the various tools available:

| Interface | Method of Definition |
|--------------------|---|
| PL/SQL | Use the DBMS_SCHEDULER.CREATE_JOB procedure using the above parameters to define the job object. |
| SQL Developer | Use the <i>New Job</i> menu item of the <i>Scheduler</i> tree to define the new job using a right click action on the <i>Jobs</i> node. |
| Enterprise Manager | Using the <i>Administration</i> → <i>Oracle Scheduler</i> → <i>Jobs</i> menu item, specify the job information as above. |

Defining Chains

Note: It is recommended to review the [Creating and Managing Job Chains](#) documentation for additional advice.

Once the jobs and schedule have been defined the next step is to define the set of job chains which configure the order and rules for a sequence of related jobs. Typically a job chain

represents a set of business related processes that must be scheduled and processed in a specific order. For example, a nightly set of jobs that constitute a set of business processes is considered a chain.

There are a various rules for a chain:

- There is a begin event and an end event that are automatically generated for a chain. The chain is considered ended when all jobs directly related to the end event are complete.
- Relationships in the chain are based upon [job state events](#) to dictate next actions. Typically, this is **SUCCEEDED** but can support other events. For example, it is possible to start the next job in a chain based upon a **FAILED** event to perform cleanup activities if required.

There are a number of key steps outlined in [Creating and Managing Job Chains](#) that should be performed with the following additions building chains that include product jobs:

- The naming conventions for the objects defined in the chain are subject to rules for all Oracle dictionary objects. It is recommended to prefix customizations with **CM_** to avoid conflicts with base provided objects.
- It is recommended to implement a chain step per batch control that is applicable to the chain. The step name can match the Batch control or a valid alternative for your implementation.
- For each sequence in the chain, set the appropriate state in a Chain Rule for the setup. For example, if StepB must start only if StepA is complete then set the condition for the rule for Step A as "StepA SUCCEEDED" and action as "START StepB". Complex rules are supported. Refer to [Adding Rules to a Chain](#) for more details.

There are number of ways that can be used to specify chains using the various tools available:

| Interface | Method of Definition |
|--------------------|---|
| PL/SQL | Refer to Creating and Managing Job Chains for the sequence of statements to execute. |
| SQL Developer | Use the <i>New Chain</i> menu item of the <i>Scheduler</i> tree to define the new chain using a right click action on the <i>Chains</i> node. |
| Enterprise Manager | Using the <i>Administration</i> → <i>Oracle Scheduler</i> → <i>Chains</i> menu item, specify the chain information. |

Other Scheduler Objects

There are a number of scheduler objects not covered in the above documentation that can be used by implementations using the following guidelines:

- [File Watchers](#) - These objects create events based upon an arrival of a file in a configured location. To use File Watchers, a destination and credentials need to be defined along with the File Watcher itself. The file needs to be accessible by the process that is using it.
- [Windows](#) - These are definitions of times and dates where certain conditions, restrictions or resource profiles can be used. Oracle Scheduler supports discrete or

- overlapping Windows²⁸.
- [Groups](#) - It is possible to group certain scheduler objects for efficient administration and clustering purposes.

Executing Jobs

There are a number of techniques that can be used to execute a schedule and/or a job. The following techniques can be used:

- Chains can be executed using a predefined schedule or manually started by starting the job created for the chain. Refer to [Running Jobs](#) for additional information.
- Events can trigger jobs when the event occurs. Refer to [Using Events to Start Jobs](#) for more information.

Chains and Jobs that are enabled are only executed.

Monitoring Oracle Scheduler

There are various techniques available from the interface and related tools to monitor Oracle Scheduler. Refer to [Monitoring Jobs](#) for additional advice.

Note: Any product job can also be monitored using the Batch Run Tree, JMX interface or via the Oracle Utilities Batch Server target page within Oracle Enterprise Manager via the Application Management Pack for Oracle Utilities.

Job States

When any job is waiting to be executed, executing, completed or errored the scheduler updates the state of the job in its job tables. These states are available from the tools mentioned earlier in various forms. The states that are raised are documented in [Monitoring Job State with Events Raised by the Scheduler](#).

Dictionary Views

The most common technique for monitoring the scheduler itself is the provided [Scheduler Data Dictionary Views](#). These views not only provide details of the Oracle Scheduler Objects but can provide execution information. There are two views that are of particular interest for executions:

- ***_SCHEDULER_RUNNING_CHAINS** - Show all chains that have active jobs.
- ***_SCHEDULER_RUNNING_JOBS** - Show state information for all jobs running at the moment.

These views are accessible from any SQL product or via Oracle Enterprise Manager via the *Administration* → *Oracle Scheduler* → *Home* menu item on the database target used by the Oracle Scheduler.

²⁸ Whilst supported, Oracle does not recommend overlapping windows.

Scheduler Notifications

One of the features of the Oracle Scheduler is that it is possible to send an email based upon the state of a particular job within the scheduler. This is based upon the state of the job and the email format can be customized based upon the state. For more information and examples of the capability refer to [Monitoring Job State with E-mail Notifications](#).

The Oracle Scheduler Interface includes the **OUAF_NOTIFY** job that can be used as a template.

Cancellation

Once a job has been executed there are two techniques for cancelling a running job or thread:

- Use the [OUAF_BATCH.Cancel_Job](#) API to stop the thread or job. The job state will be set to **STOPPED**.
- Use the [Batch JMX API](#) to cancel a thread.

The cancellation will be registered on the batch run tree including any relevant security information for auditing purposes.

Exporting Schedules across environments

As the schedule is a set of dictionary objects it can be exported and imported using any of the Oracle database copy, cloning or exporting functions. Utilities such as **exp** or **expdp** can be used to export the schema of the owner of the schedule.

It is also possible to isolate the scheduler objects using the data pump **INCLUDE** function. The **INCLUDE** statement should use the following filter:

```
INCLUDE=procobj:"IN (SELECT NAME FROM sys.OBJ$ WHERE TYPE# IN (47,48,66,67,68,69,71,72,74))"
```

For example:

```
expdp <user>/<password> DIRECTORY=<directory> DUMPFILE=<expfile>
LOGFILE=<logfile> SCHEMAS=<schema> INCLUDE=procobj:"IN (SELECT NAME
FROM sys.OBJ$ WHERE TYPE# IN (47,48,66,67,68,69,71,72,74))"
```

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 |

- 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.

ALLOW_SELF_SIGNED_SSL – enable self signed SSL

| | |
|--------------|--|
| Parameter | ALLOW_SELF_SIGNED_SSL |
| Description | Specifies if self signed SSL is allowed |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

appViewer - Name of appViewer WAR file

| | |
|--------------|---------------------------------------|
| Parameter | appviewer |
| Description | Name of AppViewer WAR file |
| Ordinality | Mandatory |
| Valid Values | This value is automatically generated |

Tier **WEB**

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]

Tier **BAS** **BATCH**

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]

Tier **BATCH**

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]

Tier **BAS** **BATCH**

BATCHTHREADS - Number of Online Batch Server Threads

Parameter **BATCHTHREADS**

Description Number of Threads in DEFAULT online threadpool.

Ordinality Mandatory

Valid Values Default: 5

Tier **BAS**

BATCH_DBPASS - Batch User Password

Parameter **BATCH_DBPASS**

Description Encrypted password for database user used for batch.

| | |
|--------------|------------------------|
| Ordinality | Mandatory |
| Valid Values | Encrypted password |
| Tier | BATCH DB |

BATCH_DBUSER - Batch User

| | |
|--------------|-------------------------------|
| Parameter | BATCH_DBUSER |
| Description | Database user used for batch. |
| Ordinality | Mandatory |
| Valid Values | Valid database user |
| Tier | BATCH DB |

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: <code>BATCH_MEMORY_ADDITIONAL_OPT=-Xdebug -Xnoagent -Xrunjdp:transport=dt_socket,address=7757,server=y,suspend=n -Djava.compiler=NONE</code> |
| Tier | BATCH |

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 |
| Tier | BATCH |

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 |
| Tier | BATCH |

BATCH_MODE - Default Mode of Batch

| | |
|--------------|--------------------------------------|
| Parameter | BATCH_MODE |
| Description | Batch Execution Mode for Threadpools |
| Ordinality | Mandatory |
| Valid Values | [THIN CLUSTERED] |
| Tier | BATCH |

BATCH_ONLINE_LOG_DIR - Online Batch Submission Log Directory

| | |
|--------------|--|
| Parameter | BATCH_ONLINE_LOG_DIR |
| Description | Location of Online Batch Submission logs |
| Ordinality | Optional |
| Valid Values | Valid directory |
| Tier | BAS |

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 |
| Tier | BATCH |

BSN_APP - Business Server Application Name

| | |
|--------------|---|
| Parameter | BSN_APP |
| Description | Deployed Business Application Server Name |
| Ordinality | Mandatory |
| Valid Values | Default: SPLService |
| Tier | BAS |

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 |
| Tier | BAS |

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 |
| Tier | BAS |

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 |
| Tier | BAS |

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 |
| Tier | BAS |

CALC_WLS_COMMON_JAR_DIR – Weblogic’s Jersey Jars directory

| | |
|-------------|---|
| Parameter | CALC_WLS_COMMON_JAR_DIR |
| Description | The full path to the jersey Jars in weblogic install directory. |

| | |
|--------------|---|
| Ordinality | Mandatory |
| Valid Values | Full path |
| Tier | WEB BAS BATCH MOB |

CALC_WLS_SERVER_JAR_DIR– Weblogic server lib jars directory

| | |
|--------------|--|
| Parameter | CALC_WLS_SERVER_JAR_DIR |
| Description | The full path to the weblogic server lib Jars in weblogic install directory. |
| Ordinality | Mandatory |
| Valid Values | Full path |
| Tier | WEB BAS BATCH MOB |

CHAR_BASED_DB – specifies if the DB is char based

| | |
|--------------|---|
| Parameter | CHAR_BASED_DB |
| Description | Specifies if the DB is char based. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS BATCH MOB |

CLOUD_CUSTOM_WHITE_LIST_PATH - Custom URL Whitelists for Cloud

| | |
|--------------|---|
| Parameter | CLOUD_CUSTOM_WHITE_LIST_PATH |
| Description | The full path to a custom whitelist for aliases for URLs within the product. This whitelist is added to the base product whitelist which is provided in CLOUD_WHITE_LIST_PATH variable. This is primarily for cloud implementations to define URL aliases . |
| Ordinality | Optional |
| Valid Values | Full path (including filename to whitelist) |
| Tier | WEB BAS BATCH MOB |

CLOUD_LOCATION_F1_BASE_IWS_URL – URI for variable F1_BASE_IWS_URI

| | |
|--------------|--|
| Parameter | CLOUD_LOCATION_F1_BASE_IWS_URI |
| Description | The base WSDL URI is different than the application server URI and must be configured for cloud implementations. Need not be configured for on-premise implementations.. |
| Ordinality | Optional |
| Valid Values | Full path |
| Tier | WEB BAS |

CLOUD_LOCATION_F1_BASE_REST_URL – URI for variable F1_BASE_REST_URL

| | |
|--------------|--|
| Parameter | CLOUD_LOCATION_F1_BASE_REST_URL |
| Description | URI for variable F1_BASE_REST_URL. This is for implementations to define the base URI for REST. This is primarily for cloud implementations.. |
| Ordinality | Optional |
| Valid Values | Full path |
| Tier | WEB BAS |

CLOUD_LOCATION_F1_BASE_WEB_URI – URI for variable F1_BASE_WEB_URI

| | |
|--------------|--|
| Parameter | CLOUD_LOCATION_F1_BASE_WEB_URI |
| Description | The environment URL used to launch the application server from an external link must be configured for cloud implementations. Need not be configured for on-premise implementations. |
| Ordinality | Optional |
| Valid Values | Full path |
| Tier | WEB BAS |

CLOUD_LOCATION_F1_MIGR_ASSISTANT_FILES – Location of CMA files for migrations

| | |
|-----------|---|
| Parameter | CLOUD_LOCATION_F1_MIGR_ASSISTANT_FILES |
|-----------|---|

| | |
|--------------|--|
| Description | The full path to the location of the CMA files for configuration migration. This is primarily for cloud implementations to define configuration file location for CMA process . |
| Ordinality | Optional |
| Valid Values | Full path |
| Tier | WEB BAS BATCH |

CLOUD_LOCATION_F1_OAUTH2_URI – URI for OAUTH2

| | |
|--------------|---|
| Parameter | CLOUD_LOCATION_F1_OAUTH2_URI |
| Description | The URI for implementing OAUTH2. This is primarily for cloud implementations to implment OAUTH2. |
| Ordinality | Optional |
| Valid Values | Full URI |
| Tier | WEB BAS BATCH |

CLOUD_RESTRICTIONS_URLS_ENABLE - Enable or Disable URL Restrictions

| | |
|--------------|--|
| Parameter | CLOUD_RESTRICTIONS_URL_ENABLE |
| Description | Whether the whitelists for URL's are enabled. This is primarily for cloud implementations. |
| Ordinality | Optional |
| Valid Values | true false |
| Tier | WEB BAS BATCH MOB |

CLOUD_SUBSTITUTION_VARIABLE_LIST_FILE_LOCATION - Enable or Disable URL Restrictions

| | |
|--------------|---|
| Parameter | CLOUD_SUBSTITUTION_VARIABLE_LIST_FILE_LOCATION |
| Description | Points to the file containing the predefined URI substitution variable list. This is primarily for cloud implementations. |
| Ordinality | Optional [Mandatory only for Cloud implementations] |
| Valid Values | Full path (including the filename for substitution list) |
| Tier | WEB BAS BATCH MOB |

CLoud_WHITE_LIST_PATH - URL Whitelists for Cloud

| | |
|--------------|--|
| Parameter | CLoud_WHITE_LIST_PATH |
| Description | The full path to a base whitelist for aliases for URLs within the product. This whitelist is added with the custom product whitelist which is provided in CLoud_CUSTOM_WHITE_LIST_PATH variable. This is primarily for cloud implementations to define URL aliases . |
| Ordinality | Optional |
| Valid Values | Full path (including filename to whitelist) |
| Tier | WEB BAS BATCH MOB |

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 |
| Tier | BAS BATCH MOB |

CM_INSTALL_SAMPLE - Whether samples are installed

| | |
|--------------|--|
| Parameter | CM_INSTALL_SAMPLE |
| Description | Whether sample code is installed or not This is done at installation time |
| Ordinality | Mandatory |
| Valid Values | true false |
| Tier | WEB BAS BATCH MOB |

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. |

Tier **BATCH**

COHERENCE_CLUSTER_MODE - Batch Cluster Mode

| | |
|--------------|---|
| Parameter | COHERENCE_CLUSTER_MODE |
| Description | <p>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.</p> <p>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</p> |
| Ordinality | Mandatory |
| Valid Values | [dev prod] |
| Tier | BATCH |

COHERENCE_CLUSTER_NAME - Batch Cluster Name

| | |
|--------------|---|
| Parameter | COHERENCE_CLUSTER_NAME |
| Description | Unique Cache name for Coherence Clusters. |
| Ordinality | Mandatory |
| Valid Values | <p>Any string value. It is recommended to use the SPLENVIRON value, optionally, with the database schema owner as a unique identifier. For example:</p> <p>DEV.SPLADM</p> |
| Tier | BATCH |

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 . |
| JEE 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. |
| Tier | MOB |

CONSOLIDATED_LOG_FILE_PATH- Directory for consolidated log file

| | |
|--------------|--|
| Parameter | CONSOLIDATED_LOG_FILE_PATH |
| Description | Specifies the directory for the consolidated log file. |
| Ordinality | Optional |
| Valid Values | Full directory path. |
| Tier | WEB BAS BATCH MOB |

CSRF_PROTECTION – Cross-Site Request Forgery protection

| | |
|--------------|--|
| Parameter | CSRF_PROTECTION |
| Description | Specifies if Cross-Site Request Forgery protection is enabled or disabled. |
| Ordinality | Mandatory |
| Valid Values | [true false]. |
| Tier | WEB BAS BATCH MOB |

CUSTOM_SQL_SECURITY – Enable custom SQL security

| | |
|--------------|--|
| Parameter | CUSTOM_SQL_SECURITY |
| Description | Specifies if custom SQL security is enabled or disabled. |
| Ordinality | Mandatory |
| Valid Values | [true false]. |
| Tier | WEB BAS BATCH MOB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

DBNAME - Database Name

| | |
|--------------|--|
| Parameter | DBNAME |
| Description | Oracle Database Instance Name. Customers using Native mode installation can ignore this value if using JDBC Data Sources. |
| Ordinality | Mandatory |
| Valid Values | Valid Database instance name |
| Tier | BAS BATCH DB |

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 |
|--------------|--------------------|

| | |
|------|-----------------------------------|
| Tier | BAS BATCH DB |
|------|-----------------------------------|

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 |
|--------------|--------------------|

| | |
|------|----------------------|
| Tier | MOB DB |
|------|----------------------|

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 |
|--------------|--------------------|

| | |
|------|-----------------------------------|
| Tier | BAS BATCH DB |
|------|-----------------------------------|

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 |
| Tier | BAS BATCH DB |

DBPASS_OSB - OSB Database Password

| | |
|--------------|---|
| Parameter | DBPASS_OSB |
| Description | <p>Encrypted database user password for the Oracle Service Bus (OSB) integration. Password for DBUSER_OSB.</p> <p>This password was specified as part of the Oracle Service Bus installation for the OSB database user.</p> <p>This is only valid for products that include Oracle Service Bus. This is used for deployment of content to Oracle Service Bus for integration.</p> <p>This is not used for the Oracle Service Bus Adapters for Oracle Utilities.</p> |
| Ordinality | Mandatory |
| Valid Values | Encrypted Password |
| Tier | BAS BATCH DB |

DBPASS_SOAINFRA - SOA Database Password

| | |
|--------------|---|
| Parameter | DBPASS_SOAINFRA |
| Description | <p>Encrypted database user password for the SOA Suite integration. Password for DBUSER_SOAINFRA.</p> <p>This password was specified as part of the SOA Suite database installation process for the <i>prefix_SOAINFRA</i> user.</p> <p>This is only valid for products that include Oracle SOA Suite. This is used for deployment of content to Oracle SOA Suite for integration.</p> |
| Ordinality | Mandatory |
| Valid Values | Encrypted Password |
| Tier | BAS BATCH DB |

DBPORT - Database Listener Port

| | |
|-------------|---|
| Parameter | DBPORT |
| Description | <p>Oracle Listener port for connection to the database.</p> <p>Customers using Native mode installation can ignore this value if using JDBC Data Sources.</p> |

| | |
|--------------|-----------------------------------|
| Ordinality | Mandatory |
| Valid Values | Valid Port Number |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 <i>prefix_MDS</i> 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 (<i>prefix_MDS</i>) on target database |
| Tier | BAS BATCH DB |

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 (<i>prefix_ORASDPM</i>) on target database |
| Tier | BAS BATCH DB |

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 |
| Tier | BAS BATCH DB |

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 <i>prefix_SOAINFRA</i> 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 (<i>prefix_SOAINFRA</i>) on target database |
| Tier | BAS BATCH DB |

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. |
| Tier | BAS BATCH DB |

DESC - Environment Description

| | |
|--------------|--|
| Parameter | DESC |
| Description | Environment Description. Used for documentation purposes only. |
| Ordinality | Optional |
| Valid Values | String Value |
| Tier | 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 |
| Tier | BAS BATCH DB |

DIRSEP - Directory separator

| | |
|-------------|---|
| Parameter | DIRSEP |
| Description | Directory Separator 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 Separator character |
| Tier | WEB BAS IWS BATCH MOB |

DOC1BILLSSCRIPT - Name Of External Print Script

| | |
|--------------|---|
| Parameter | DOC1BILLSSCRIPT |
| 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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

ENABLE_JMS_GLOBAL_FLUSH – Enable Flusing of JMS

| | |
|--------------|--|
| Parameter | ENABLE_JMS_GLOBAL_FLUSH |
| Description | This specifies if the global JMS flusing is enabled. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS IWS BATCH MOB |

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] |
| Tier | WEB BAS IWS BATCH MOB |

ENVIRONMENT_ID - Environment Identifier

| | |
|--------------|---|
| Parameter | ENVIRONMENT_ID |
| Description | <p>Unique identifier for each environment or copy of the product. This is specified at installation time to link disparate installations into a single environment for management purposes.</p> <p>This setting is used by the Oracle Application Management Pack for Oracle Utilities within Oracle Enterprise Manager to link disparate targets across a number of hosts into a single Oracle Utilities Environment target.</p> |
| Ordinality | Mandatory |
| Valid Values | Autogenerated |
| Tier | WEB BAS IWS BATCH |

FORCE_HTTP – Enforce http protocol

| | |
|--------------|--|
| Parameter | FORCE_HTTP |
| Description | Specifies if http protocol needs to be enforced. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS IWS MOB BATCH |

FW_VERSION - Oracle Utilities Application Framework version

| | |
|--------------|--|
| Parameter | FWVERSION |
| Description | <p>Oracle Utilities Application Version</p> <p>This is an internal setting that is generated upon installation of the product, service pack installation or upgrade installation.</p> <p>This setting is automatically generated at installation time and should not be altered.</p> |
| Ordinality | Mandatory |
| Valid Values | Valid Oracle Utilities Application Framework version |
| Tier | Not Applicable |

FW_VERSION_NUM - Oracle Utilities Application Framework Version Identifier

| | |
|--------------|--|
| Parameter | FW_VERSION_NUM |
| Description | <p>Oracle Utilities Application Version</p> <p>This is an internal setting that is generated upon installation of the product, service pack installation or upgrade installation.</p> <p>This setting is automatically generated at installation time and should not be altered.</p> |
| Ordinality | Mandatory |
| Valid Values | Valid Oracle Utilities Application Framework version |
| Tier | Not Applicable |

GIS - GIS Support

| | |
|--------------|--|
| Parameter | GIS |
| Description | <p>Enables or Disables GIS support within the product.</p> <p>This setting is only used by products that use GIS integration.</p> <p>Setting this value to true enabled all other GIS settings.</p> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS BATCH |

GIS_URL - GIS Service URL

| | |
|--------------|---|
| Parameter | GIS_URL |
| Description | <p>Specifies the GIS JNDI URL used for connections.</p> <p>This setting is only used by products that use GIS integration.</p> <p>This setting is only used if GIS is set to true.</p> |
| Ordinality | Mandatory |
| Valid Values | Valid JNDI URL |
| Tier | WEB BAS BATCH |

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 |
| Tier | WEB BAS BATCH |

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 |
| Tier | WEB BAS BATCH |

GLOBAL_JVMARGS - Additional JVM Arguments

| | |
|--------------|--|
| Parameter | GLOBAL_JVMARGS |
| Description | Specific additional JVM options for the online component of the product for Oracle JVM (to enable advanced features) |
| Ordinality | Mandatory |
| Valid Values | This is generated |
| Tier | WEB BAS |

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. |
| Tier | BAS |

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. |
| Tier | BAS |

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. |
| Tier | BAS BATCH IWS |

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. |
| Tier | WEB BAS IWS BATCH MOB |

HSTS_MAX_AGE – Maximum age of strict transport security

| | |
|-------------|---|
| Parameter | HSTS_MAX_AGE |
| Description | Maximum age of strict transport security. |

| | |
|--------------|--------------------------------------|
| Ordinality | Optional. Default Value is 31536000. |
| Valid Values | Value in seconds. |
| Tier | WEB |

HSTS_PRELOAD – Strict transport security preload

| | |
|--------------|---|
| Parameter | HSTS_PRELOAD |
| Description | Preload of strict transport security. |
| Ordinality | Optional |
| Valid Values | false true. Default values is false. |
| Tier | WEB |

HSTS_SUBDOMAINS – Subdomains included in strict transport security

| | |
|--------------|--|
| Parameter | HSTS_SUBDOMAINS |
| Description | Subdomains that are included in strict transport security. |
| Ordinality | Optional |
| Valid Values | false true. Default value is true. |
| Tier | WEB |

IWS_JDBC_NAME - JNDI Name for the IWS Connection pools

| | |
|--------------|--|
| Parameter | IWS_JDBC_NAME |
| Description | This is the JDBC Data Source connection name used for Inbound Web Services. The JDBC Data Source should be created prior to setting this value using the Configuring JDBC Data Sources instructions. |
| Ordinality | Optional |
| Valid Values | Valid fully qualified JNDI name for JDBC Data Source. |
| Tier | IWS |

IPCSTARTPORT - Starting IPC Port for Scheduler

| | |
|-------------|--|
| Parameter | IPCSTARTPORT |
| Description | Starting port number for Interprocess Communications (IPC) for Scheduler |
| Ordinality | Mandatory |

| | |
|--------------|-------------------------|
| Valid Values | Valid open port numbers |
| Tier | MOB |

IWSWAR - Name of Inbound Web Services WAR

| | |
|--------------|---|
| Parameter | IWSWAR |
| Description | Default Name for Inbound Web Services Application Name. |
| Ordinality | Mandatory |
| Valid Values | Valid Server Application Names |
| Tier | IWS |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

JAVA_VENDOR - Java Vendor

| | |
|--------------|--|
| Parameter | JAVA_VENDOR |
| Description | JVM Vendor |
| Ordinality | Mandatory |
| Valid Values | Generated and used in other utilities |
| Tier | WEB BAS IWS BATCH MOB |

JAVA_VERSION - JVM Version string

| | |
|--------------|--|
| Parameter | JAVA_VERSION |
| Description | Fully qualified JVM version string |
| Ordinality | Mandatory |
| Valid Values | Generated and used in other utilities |
| Tier | WEB BAS IWS BATCH MOB |

JAVA_VERSION_MAIN - Abberviated JVM Version

| | |
|--------------|--|
| Parameter | JAVA_VERSION_MAIN |
| Description | Abbreviated JVM version string |
| Ordinality | Mandatory |
| Valid Values | Generated and used in other utilities |
| Tier | WEB BAS IWS BATCH MOB |

JDBC_NAME - JNDI Name for the Online JDBC Data Source

| | |
|--------------|--|
| Parameter | JDBC_NAME |
| Description | This is the JDBC Data Source connection name used for Online channel. The JDBC Data Source should be created prior to setting this value using the Configuring JDBC Data Sources instructions. |
| Ordinality | Optional |
| Valid Values | Valid fully qualified JNDI name for JDBC Data Source. |
| Tier | WEB |

JDBC_TIMEOUT – JDBC read timeout

| | |
|--------------|---|
| Parameter | JDBC_TIMEOUT |
| Description | Disable Cross Site Request Forgery token validation for REST Services. The JDBC Data Source should be created prior to setting this value using the Configuring JDBC Data Sources instructions. |
| Ordinality | Optional |
| Valid Values | The value should be in milliseconds,the default is 600000 (10 minutes). |

Tier **WEB**

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 |
| Tier | BAS BATCH MOB |

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 |
| Tier | BAS BATCH |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

KS_IMPORT_KEYSTORE_FOLDER - Keystore Import Folder

| | |
|--------------|--|
| Parameter | KS_IMPORT_KEYSTORE_FOLDER |
| Description | Keystore import folder. |
| Ordinality | Mandatory |
| Valid Values | Valid location for keystore folder . |
| Tier | WEB BAS IWS BATCH MOB |

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 . <u>Linux:</u> \$SPLEBASE/ks/.ouaf_keystore <u>Windows:</u> %SPLEBASE%\ks\.ouaf_keystore |
| Tier | WEB BAS IWS BATCH MOB |

KS_MODE - Key Padding Mode

| | |
|--------------|--|
| Parameter | KS_MODE |
| Description | Key Padding Mode |
| Ordinality | Mandatory |
| Valid Values | Valid Padding for Keystore. Default: CBC |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | WEB BAS IWS BATCH MOB |

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 . <u>Linux:</u> \$SPLEBASE/ks/.ouaf_storepass <u>Windows:</u> %SPLEBASE%\ks\.ouaf_storepass |
| Tier | WEB BAS IWS BATCH MOB |

KS_STORETYPE - Key Store Type

| | |
|--------------|--|
| Parameter | KS_STORETYPE |
| Description | Keystore Store Type used by keytool . |
| Ordinality | Mandatory |
| Valid Values | Valid Key Store Type . Default: |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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 | Fully qualified directory holding map files |
| Tier | MOB |

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 |
| Tier | MOB |

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] |
| Tier | MOB |

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) |
| Tier | MOB |

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) |
| Tier | MOB |

MOBILE_APPDIR – Directory of the mobile application install

| | |
|--------------|---|
| Parameter | MOBILE_APPDIR |
| Description | Directory of the mobile application install on the server This parameter only applies to products that use the Mobile. |
| Ordinality | Optional |
| Valid Values | Full directory path. If not provided then default value is set to '<SPLOUTPUT>/mobile' |
| Tier | MOB |

MOBILE_APP_ONLY - Deploy Only Mobility Web Application

| | |
|--------------|--|
| Parameter | MOBILE_APP_ONLY |
| Description | Whether the mobile web application is the only deployment on the server. This allows for a distributed installation. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | MOB |

MOBILE_ENABLED – If Mobile application is enabled

| | |
|--------------|---|
| Parameter | MOBILE_ENABLED |
| Description | Specified if the mobile application is enabled. It enables the mobile functionalities in the initialSetup.sh process, which includes: (1) Copying the Framework mobile sources to the MOBILE_APPDIR; (2) Running the F1-BMCOM batch job; (3) Adding the MobileApp.war into the SPLWeb.ear. |
| Ordinality | Mandatory |
| Valid Values | [true false]. |
| Tier | MOB |

MODULES - Names of Modules installed (Deprecated)

| | |
|-----------|----------------|
| 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 | Calculated inside processTemplate.plx. |
| Tier | WEB BAS IWS BATCH MOB |

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 |
| Tier | BAS IWS BATCH DB |

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 |
| Tier | MOB |

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 |
| Tier | BAS BATCH |

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 |
| Tier | BAS BATCH DB |

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. |
| Tier | BAS BATCH DB |

OPSYS - Operating System

| | |
|--------------|--|
| Parameter | OPSYS |
| Description | Internal variable used for platform specific configurations to be implemented. |
| Ordinality | Mandatory |
| Valid Values | Generated Operating System String |
| Tier | WEB BAS IWS BATCH MOB |

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 . |
| Tier | WEB BAS IWS BATCH MOB |

OSB_HOME - Oracle Service Bus Home

| | |
|--------------|---|
| Parameter | OSB_HOME |
| Description | Location of 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 | Location of Oracle Service Bus installation. |
| Tier | BAS |

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. |
| Tier | BAS |

OSB_LOG_DIR - Default Location For Oracle Service Bus Logs

| | |
|--------------|---|
| Parameter | OSB_LOG_DIR |
| Description | Logging 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. |
| Tier | BAS |

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. |
| Tier | BAS |

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. |
| Tier | BAS |

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. |
| Tier | BAS |

OUAF_DBMS_SCHEDULER_USER – DB User of DBMS scheduler

| | |
|--------------|-----------------------------------|
| Parameter | OUAF_DBMS_SCHEDULER_USER |
| Description | Database user for DBMS scheduler. |
| Ordinality | Mandatory |
| Valid Values | User Name. |

Tier **BATCH**

OWSM_ENABLED – Enable Oracle Web Services Manager

| | |
|--------------|--|
| Parameter | OWSM_ENABLED |
| Description | Specifies if Oracle Web Services Manager is enabled or disabled. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS IWS |

OWSM_PROTECTION_FOR_REST_SERVICES – Enable Oracle Web Services Manager protection for REST

| | |
|--------------|---|
| Parameter | OWSM_PROTECTION_FOR_REST_SERVICES |
| Description | Specifies if Oracle Web Services Manager protection is enabled or disabled for REST services. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB BAS IWS |

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. |
| Tier | WEB BAS IWS BATCH MOB |

SERVER_ROLES - Which server roles are enabled

| | |
|--------------|---|
| Parameter | SERVER_ROLES |
| Description | Which server roles are enabled for this installation. |
| Ordinality | Mandatory |
| Valid Values | Install generated |
| Tier | WEB IWS BATCH |

SERVER_ROLE_BATCH - Whether the Batch Server is enabled in this installation

| | |
|--------------|--|
| Parameter | SERVER_ROLE_BATCH |
| Description | Whether this installation is to be used for batch processing |
| Ordinality | Mandatory |
| Valid Values | Install generated with true or false |
| Tier | BATCH |

SERVER_ROLE_INTEGRATION - Whether the IWS Server is enabled in this installation

| | |
|--------------|---|
| Parameter | SERVER_ROLE_INTEGRATION |
| Description | Whether this installation is to be used for integration (IWS) |
| Ordinality | Mandatory |
| Valid Values | Install generated with true or false |
| Tier | IWS |

SERVER_ROLE_ONLINE - Whether the online server is enabled in this installation

| | |
|--------------|--|
| Parameter | SERVER_ROLE_ONLINE |
| Description | Whether this installation is to be used for online |
| Ordinality | Mandatory |
| Valid Values | Install generated with true or false |
| Tier | WEB |

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. |
| Tier | BAS |

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. |
| Tier | BAS |

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. |
| Tier | BAS |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

SPLOUTPUT - Directory for product output

| | |
|--------------|---|
| Parameter | SPLOUTPUT |
| Description | Location of product output. |
| Ordinality | Mandatory |
| Valid Values | Valid Location of output. Generated by installer. |

Tier **WEB** **BAS** **IWS** **BATCH** **MOB**

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 |
| Tier | BAS |

SPLWEBAPP - Name of Web Application Server Application

| | |
|--------------|--|
| Parameter | SPLWEBAPP |
| Description | Default Name for Web Application Application Name. |
| Ordinality | Mandatory for Oracle WebLogic implementations |
| Valid Values | Valid Server Application Names |
| Tier | WEB MOB |

STRICT_TRANSPORT_SECURITY – Enable Strict Transport Security

| | |
|--------------|---|
| Parameter | STRICT_TRANSPORT_SECURITY |
| Description | Enable or Disable transport layer security. |
| Ordinality | Mandatory |
| Valid Values | [true false]. |
| Tier | WEB |

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] |
| Tier | BAS |

TEMPSTORAGE_SPLITFILESIZE - Split File Size in MB

| | |
|--------------|--|
| Parameter | TEMPSTORAGE_SPLITFILESIZE |
| Description | The size of split files in MB, while transferring a file using file adapter API. |
| Ordinality | Optional |
| Valid Values | Valid file size in MB. Default is 50. |
| Tier | BAS |

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. |
| Tier | MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | WEB BAS IWS BATCH MOB |

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. |
| Tier | MOB |

WEB_ADMIN_SERVER – Name of the Web Admin Server

| | |
|--------------|---|
| Parameter | WEB_ADMIN_SERVER |
| Description | Name of the web admin server. |
| Ordinality | Mandatory |
| Valid Values | Server Name |
| Tier | WEB BAS IWS MOB |

WEB_APP - Name of WebSphere Server Application

| | |
|--------------|--|
| Parameter | WEB_APP |
| Description | Default Name for Web Application Application Name. |
| Ordinality | Mandatory for WebSphere implementations |
| Valid Values | Valid Server Application Names |
| Tier | WEB MOB |

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 |
| Tier | WEB |

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 |
| Tier | WEB |

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 |
| Tier | WEB |

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 |
| Tier | WEB |
| Installation Mode | 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 |
| Tier | MOB |

WEB_CONTEXT_ROOT - Web Context Root

| | |
|--------------|---|
| Parameter | WEB_CONTEXT_ROOT |
| Description | Web Context for product |
| Ordinality | Mandatory |
| Valid Values | Valid context. Default: spl |
| Tier | WEB BAS IWS MOB |

WEB_DEPLOY_APPVIEWER - Deploy AppViewer or not

| | |
|--------------|--|
| Parameter | WEB_DEPLOY_APPVIEWER |
| Description | Whether the AppViewer application is included in the online EAR file. A value of true adds the appViewer to the online EAR file. A value of false does not add the appViewer content to the online EAR file. <i>Note: If the appViewer is not deployed the online administration functions that use the appViewer will not display this content.</i> It is recommended to set this value to false for non-development environments. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB |

WEB_ENABLE_HEALTHCHECK - Enable System Healthcheck

| | |
|--------------|--|
| Parameter | WEB_ENABLE_HEALTHCHECK |
| Description | Enable the system healthcheck function used by the Oracle Application Management Pack for Oracle Utilities available from Oracle Enterprise Manager. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Tier | WEB |

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 |
| Tier | WEB |

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 |

| | |
|------|------------|
| Tier | WEB |
|------|------------|

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.

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 |
| Tier | WEB |

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 |
| Tier | WEB |

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] |
| Tier | WEB |

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] |
| Tier | WEB BAS IWS BATCH MOB |

WEB_IWS_MDB_RUNAS_USER - Default User for MDB to run as within product

| | |
|--------------|---|
| Parameter | WEB_IWS_MDB_RUNAS_USER |
| Description | The default product user the MDB uses to run its interface. This value is used if no JMS credentials are provided on the message header. A blank value, the default, for this parameter disables the function and assumes all JMS messages will include their credentials in the head of the message. |
| Ordinality | Optional |
| Valid Values | Blank (not enabled) or a valid product authorization userid (not login id) |
| Tier | IWS |

WEB_IWS_SUPER_USERS - Users to run as in IWS

| | |
|--------------|--|
| Parameter | WEB_IWS_SUPER_USERS |
| Description | Default product user used for authorization for Inbound Web Services. This value is used for internal calls used for Inbound Web Services. |
| Ordinality | Optional |
| Valid Values | Blank (not enabled) or a valid product authorization userid (not login id) |
| Tier | IWS |

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. |
| Tier | WEB |

WEB_L2_CACHE_MODE - Turn off L2 Cache for Online Submission

| | |
|--------------|---|
| Parameter | WEB_L2_CACHE_MODE |
| Description | Enables (on) or disables (off) Level 2 cache for online submission. Used for development purposes where online submission cache is off. |
| | <i>Note: Some batch processes require L2 Cache to be disabled. Refer to individual job documentation for details.</i> |
| Ordinality | Mandatory |
| Valid Values | [on off] |
| Tier | WEB BATCH |

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. |
| Tier | WEB |

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. |
| Tier | WEB |

WEB_MOBILE_FORM_LOGIN_ERROR_PAGE - Default Mobile Form Login Error Page

| | |
|--------------|---|
| Parameter | WEB_MOBILE_FORM_LOGIN_ERROR_PAGE |
| Description | Error Page used for invalid Login for Mobile. |
| Ordinality | Mandatory |
| Valid Values | Valid JSP for error pages. The default is "/error". |
| Tier | MOB |

WEB_MOBILE_FORM_LOGIN_PAGE - Default Form Login Page

| | |
|--------------|---|
| Parameter | WEB_MOBILE_FORM_LOGIN_PAGE |
| Description | Login Page used for Mobile Login. |
| Ordinality | Mandatory |
| Valid Values | Valid JSP for login. The default is "/login". |
| Tier | MOB |

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] |
| Tier | WEB |

WEB_PRINCIPAL_NAME - Default JEE Authorization Principal

| | |
|--------------|--|
| Parameter | WEB_PRINCIPAL_NAME |
| Description | Principal Name for online application. |
| Ordinality | Mandatory |
| Valid Values | Valid Principal Name. Default: cisusers |
| Tier | WEB BAS IWS |

WEB_ROLE_NAME - Default JEE Authorization Role

| | |
|--------------|---|
| Parameter | WEB_ROLE_NAME |
| Description | Role Name for online application |
| Ordinality | Mandatory |
| Valid Values | Valid Role Name. Default: cisusers |
| Tier | WEB BAS IWS |

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 value: WL_HOME - Oracle WebLogic |
| Ordinality | Mandatory |
| Valid Values | Home location of Web Application Server software |
| Tier | WEB BAS IWS |

WEB_SERVER_VERSION - Web Server Version

| | |
|--------------|---|
| Parameter | WEB_SERVER_VERSION |
| Description | Full Version of JEE Web Application Server. Used for utilities to ensure correct templates are used. |
| Ordinality | Mandatory |
| Valid Values | Generated by installer. |
| Tier | WEB BAS IWS |

WEB_SERVER_VERSION_MAIN -

| | |
|--------------|--|
| Parameter | WEB_SERVER_VERSION_MAIN |
| Description | Substring Version of JEE Web Application Server. Used for utilities to ensure correct templates are used. |
| Ordinality | Mandatory |
| Valid Values | Generated by installer. |
| Tier | WEB BAS IWS |

WEB_WLAUTHMETHOD - Authentication Method

| | |
|--------------|---|
| Parameter | WEB_WLAUTHMETHOD |
| Description | JEE Authorization Method implemented. |
| Ordinality | Mandatory |
| Valid Values | Valid Values: <ul style="list-style-type: none"> • FORM (Default) - Use login forms • BASIC - Use Operating System Login • CLIENT-CERT - Client Certificate |
| Tier | WEB BAS IWS MOB |

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 |
| Tier | WEB |

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 |
| Tier | WEB |

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 |
| Tier | WEB |

WEB_WLSYSPASS - Oracle WebLogic JNDI System Password

| | |
|--------------|---|
| Parameter | WEB_WLSYSPASS |
| 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. |
| Tier | WEB |

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. |
| Tier | WEB |

WLS_CLUSTER_NAME - Name of Cluster for Product

| | |
|--------------|---|
| Parameter | WLS_CLUSTER_NAME |
| Description | Oracle WebLogic Cluster Name for clustered environments. This value must be set for customers using the Application Management Pack for Oracle Utilities to link the cluster target with the product. |
| Ordinality | Optional |
| Valid Values | Valid Cluster Name. |
| Tier | WEB |

WLTHININT3CLIENT_JAR_DIR - Location of Thin WebLogic T3 Client

| | |
|--------------|---|
| Parameter | WLTHININT3CLIENT_JAR_DIR |
| Description | Location of Oracle WebLogic thin Client for JMX and other interfaces. Used for integration to Oracle products for monitoring. |
| Ordinality | Optional |
| Valid Values | Valid location of wlthint3client.jar |
| Tier | IWS WEB |

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 |
| JEE Web Application Server | WLS |

| | |
|-------------------|---------------|
| Tier | WEB |
| Installation Mode | NATIVE |

XQUERIES_TO_CACHE - Size of XQuery Cache

| | |
|--------------|---|
| Parameter | XQUERIES_TO_CACHE |
| Description | Size of XQuery/XPath Cache in number of queries. Increasing the cache size may increase memory requirements for the product. This sets the value of com.oracle.XPath.LRUSize in the spl.properties file |
| Ordinality | Optional |
| Valid Values | Defaults from templates (-1 - Unlimited in size) |
| Tier | BAS BATCH IWS MOB |

XQUERY_CACHE_FLUSH_TIMEOUT - Cache Life for XQueries

| | |
|--------------|---|
| Parameter | XQUERY_CACHE_FLUSH_TIMEOUT |
| Description | Length of time in seconds that XQuery and XPath queries are cached and set the value of com.oracle.XPath.flushTimeout in the spl.properties file. |
| Ordinality | Optional |
| Valid Values | Defaults from templates (0 - Never flush is the default) |
| Tier | BAS BATCH IWS MOB |

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 online and web Services JEE applications for the product are controlled by this configuration file which holds the behavioral configuration for the applications. Refer to <http://jcp.org/en/jsr/detail?id=109> for more details of the XML format.

The following sections outline the specific settings for the product.

Filter Mappings - JEE Filters

Note: Filters should not be altered unless instructed by Oracle Support. Refer to the web.xml file for the relevant settings.

The following filters apply to the product:

| Filter | Usage |
|--------------------------------------|--|
| Browser Cache Filter | Browser Client side caching settings - Used with MaxAge . |
| CSS Stylesheet Direction Filter | CSS Filter |
| Doctype Replace Filter | JDOC Filter |
| FlushAll Redirect Filter | Cache management utility Filter |
| GZip Compression Filter | HTTP 1.1 compression filter |
| Header Security Filter | OWASP processing of HTTP headers using SAMEORIGIN |
| HTML Comment stripping Filter | Removal of comments from HTTP transmission |
| Image Browser Cache Filter | Browser Client side caching settings for image files - Used with MaxAgeI . |
| Image RTL Filter | Right to Left support Filter |
| Javascript Comment stripping Filter | Removal of comments from javascript transmission |
| Minification Filter | Optimization code |
| Privacy Page Redirect Filter | Privacy page redirect to /cm/privacy.html |
| Request Context Filter | Generic context Filter |
| Rest Security Filter | REST security support |
| Security Filter | Generic security filter to delegate security to JEE container |
| Timezone Database ContentType Filter | Time zone support |

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>

```

maxAge - 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 JEE 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: | <pre> <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> </pre> |

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 JEE 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: | <pre> <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> </pre> |

auth-method - Authorization Method

| | |
|---------------|--|
| Parameter | auth-method |
| Description | Sets the JEE 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. <ul style="list-style-type: none"> • 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. • DIGEST - Use digest authentication²⁹. • FORM - This instructs the product to implement the following additional settings: <ul style="list-style-type: none"> ▪ 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. <p><i>Note: Appviewer must be enabled for this setting to apply.</i></p> |
| Applicability | WEB |
| Example: | <pre><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></pre> |

²⁹ Not supported for an external LDAP provider or custom provider within Oracle WebLogic. Digest Authentication is only supported if SIP is installed with Oracle WebLogic.

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: | <pre><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></pre> |

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: | <pre><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></pre> |

spl.properties

Threadpool **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>

Note: Configuration settings having the letters 'mwm.' in their naming are applicable only to the MWM application. Example: spl.mwm.abr.contextFactory.

calendar.japanese.eras - Japanese Era Definition

| | |
|---------------|--|
| Parameter | calendar.japanese.eras |
| Description | This is a set of strings defining the era used in the Japanese calendar. The string is a collection of the following tags (one set per era separated with a ";" delimiter): <ul style="list-style-type: none"> • name - Name of era • abbr - Era Abberviation • since - Date era starts in ISO format (YYYY-MM-DD) • endDate - Date era ends in ISO format (YYYY-MM-DD) • japAbbr - The Japanese character for the era in Unicode format <p><i>Note: This feature is only appropriate if the Japanese language pack is also installed.</i></p> |
| Ordinality | Mandatory |
| Valid Values | String set by implementations |
| Source | Value inherited from templates. Use custom template to set this value, if desired. |
| Applicability | BAS BATCH |
| Example: | <code>calendar.japanese.eras=name:Taisho,abbr:T,since:1912-07-30,endDate:1926-12-24,japAbbr:\u5927\u6B63;name:Showa,abbr:S,since:1926-12-25,endDate:1989-01-07,japAbbr:\u662D\u548C;name:Heisei,abbr:H,since:1989-01-08,endDate:,japAbbr:\u5E73\u6210</code> |

com.oracle.ouaf.customer.whitelist.file – Customer's URL whitelist file

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.customer.whitelist.file |
| Description | Full directory path to the URL customer's whitelist file to be checked for restricting URLs. |
| Ordinality | Optional |
| Valid Values | Location of whitelist URL file, along with filename. |
| Source | Defaulted from the CLOUD_CUSTOM_WHITE_LIST_PATH value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.customer.whitelist.file=@SPLEBASE@/etc/cm_witelist.xml</code> The installed product base directory is defaulted from SPLEBASE value in ENVIRON.INI |

com.oracle.ouaf.consolidatedLog.fileName – Consolidate log file

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.consolidatedLog.fileName |
| Description | Full path to the consolidated log file to retrieve logs in clustered cloud environments. |
| Ordinality | Optional |
| Valid Values | Location of consolidate log file, along with filename. |
| Source | Defaulted from the CONSOLIDATED_LOG_DIR value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.consolidatedLog.fileName=@SPLEBASE@/etc/cm_whitelist.xml.</code> The installed product base directory is defaulted from SPLEBASE value in ENVIRON.INI |

com.oracle.ouaf.dbms.scheduler.schema – DBMS Scheduler object's schema

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.dbms.scheduler.schema |
| Description | The DBMS scheduler objects will be created in the user/schema specified by the property. This property should match the schema where the DBMS integration package was installed in. |
| Ordinality | Optional |
| Valid Values | DB Schema name where DBMS integration package is installed. |
| Source | Defaulted from the value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.dbms.scheduler.schema=DBSchemaName</code> |

com.oracle.ouaf.file.tempstorage.dir – location for temporary file

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.file.tempstorage.dir |
| Description | The file being transferred using the file adapter API of OUAF will be split into multiple files and stored temporarily in this location. |
| Ordinality | Optional |
| Valid Values | Directory path on the server. |
| Source | Defaulted from the value in ENVIRON.INI . |
| Applicability | BAS BATCH |
| Example: | <code>com.oracle.ouaf.file.tempstorage.dir=/ouaf/sp1/tempdirector y</code> |

com.oracle.ouaf.file.tempstorage.splitfilesize – split file size in MB

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.file.tempstorage.splitfilesize |
| Description | The file being transferred using the file adapter API of OUAF will be split into multiple files based on the size specified against this property. |
| Ordinality | Optional |
| Valid Values | Size in MB. Default is 50. |
| Source | Defaulted from the value in ENVIRON.INI . |
| Applicability | BAS BATCH |
| Example: | com.oracle.ouaf.file.tempstorage.splitfilesize=75 |

com.oracle.ouaf.migrationAssistant.exportDirectory – Directory for exported CMA file

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.migrationAssistant.exportDirectory |
| Description | Contains a directory specification, using the defined substitution variable @F1_MIGR_ASSISTANT_FILES@, in the white list. |
| Ordinality | Optional |
| Valid Values | Location of the CMA export file on the server. |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS BATCH |
| Example: | com.oracle.ouaf.migrationAssistant.exportDirectory=@F1_MIGR_ASSISTANT_FILES@/export |

com.oracle.ouaf.migrationAssistant.importDirectory – Directory for CMA file for import

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.migrationAssistant.importDirectory |
| Description | Contains a directory specification, using the defined substitution variable @F1_MIGR_ASSISTANT_FILES@, in the white list. |
| Ordinality | Optional |
| Valid Values | Location of the CMA import file on the server. |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS BATCH |
| Example: | com.oracle.ouaf.migrationAssistant.importDirectory=@F1_MIGR_ASSISTANT_FILES@/import |

com.oracle.ouaf.migrationAssistant.fileExtension – Extension of CMA file

| | |
|-----------|---|
| Parameter | com.oracle.ouaf.migrationAssistant.fileExtension |
|-----------|---|

| | |
|---------------|--|
| Description | Specifies the file extension of the CMA file. |
| Ordinality | Optional |
| Valid Values | File Extensions. |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS BATCH |
| Example: | <code>com.oracle.ouaf.migrationAssistant.fileExtension=cma</code> |

com.oracle.ouaf.mobile.appdir – directory for mobile application

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.mobile.appdir |
| Description | Specifies the directory for server component of the mobile application. |
| Ordinality | Optional |
| Valid Values | Directory path. |
| Source | If MOBILE_APPDIR is defined then it is used else, defaults to <%SPLOUTPUT%/mobile. |
| Applicability | MOB |
| Example: | <code>calculated value.</code> |

com.oracle.ouaf.proxy.password– Proxy credentials for senders if JVM is configured to use proxy

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.proxy.password |
| Description | Password corresponding to a username set using 'com.oracle.ouaf.proxy.user' property to utilize for senders when the jvm is configured to utilize proxy via variables like http.proxyHost and http.proxyPort. |
| Ordinality | Optional |
| Valid Values | Encrypted value obtained using keystore. |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS BATCH |
| Example: | <code>com.oracle.ouaf.proxy.password =xds12312SA!@sdf</code> |

com.oracle.ouaf.proxy.user – Proxy credentials for senders if JVM is configured to use proxy

| | |
|-----------|-----------------------------------|
| Parameter | com.oracle.ouaf.proxy.user |
|-----------|-----------------------------------|

| | |
|---------------|---|
| Description | Plain text user name to utilize for senders when the jvm is configured to utilize proxy via variables like http.proxyHost and http.proxyPort. |
| Ordinality | Optional |
| Valid Values | Plain text user name. |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS BATCH |
| Example: | <code>com.oracle.ouaf.proxy.user=user1</code> |

com.oracle.ouaf.system.database.disableTagging - Disables Database tagging

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.system.database.disableTagging |
| Description | This enables (false) or disables (true) the database connection tagging feature. <i>Note: Disabling this feature can reduce monitoring capabilities, disable support for Oracle Audit Vault, disable session resource management in Oracle Real Application Clusters and disable advanced support for Database resource plans.</i> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to true , if desired. |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.database.disableTagging=false</code> |

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 | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.keystore.alias=ouaf.system</code> |

com.oracle.ouaf.system.keystore.alias_keyalg - Keystore Alias Algorithm

| | |
|-----------|---|
| Parameter | com.oracle.ouaf.system.keystore.alias_keyalg |
|-----------|---|

| | |
|---------------|--|
| Description | Key Alias Algorithm used by keystore used by <code>keytool</code> . |
| Ordinality | Mandatory |
| Valid Values | Set to AES by default |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.keystore.alias_keyalg=AES</code> |

com.oracle.ouaf.system.keystore.alias_keysize – Keystore Keysize Strength

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.system.keystore.alias_keysize |
| Description | Keysize strength used by keystore used by <code>keytool</code> |
| Ordinality | Mandatory |
| Valid Values | Set to 128 by default |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.keystore.alias_keysize=128</code> |

com.oracle.ouaf.system.keystore.hmac_alias_keyalg - HMAC Algorithm

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.system.keystore.hmac_alias_keyalg |
| Description | Key HMAC Alias Algorithm used by keystore used by <code>keytool</code> . |
| Ordinality | Mandatory |
| Valid Values | Set to HmacSHA256 by default. |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.keystore.hmac_alias_keyalg=HmacSHA256</code> |

com.oracle.ouaf.system.keystore.hmac_alias_keysize - HMAC Key Size

| | |
|--------------|--|
| Parameter | com.oracle.ouaf.system.keystore.hmac_alias_keysize |
| Description | HMAC key size used by keystore used by <code>keytool</code> . |
| Ordinality | Mandatory |
| Valid Values | Set to 256 by default. |
| Source | Value inherited from templates. Use custom template to set this value. |

(user exit: FW_spl.properties.keystore.truststore.include).

Applicability **WEB** **BAS** **IWS** **BATCH** **MOB**
 Example: `com.oracle.ouaf.system.keystore.hmac_alias_keysize=256`

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** **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**) Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include).
 Applicability **WEB** **BAS** **IWS** **BATCH** **MOB**
 Example: `com.oracle.ouaf.system.keystore.file=/opt/.ouaf_keystore`

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 Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include).

Applicability **WEB** **BAS** **IWS** **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 Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include).
 Applicability **WEB** **BAS** **IWS** **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**) Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include).
 Applicability **WEB** **BAS** **IWS** **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 **PKCS12**.
 Source Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include).
 Applicability **WEB** **BAS** **IWS** **BATCH** **MOB**

Example: `com.oracle.ouaf.system.keystore.type=PKCS12`

com.oracle.ouaf.system.truststore.alias - Truststore Alias

| | |
|---------------|--|
| Parameter | <code>com.oracle.ouaf.system.truststore.alias</code> |
| Description | Sets name alias for truststore. Corresponds to the truststore option on keytool . |
| Ordinality | Mandatory |
| Valid Values | Set to ouaf.system . |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.truststore.alias=ouaf.system</code> |

com.oracle.ouaf.system.truststore.alias_keyalg - Truststore Key Algorithm

| | |
|---------------|--|
| Parameter | <code>com.oracle.ouaf.system.truststore.alias_keyalg</code> |
| Description | Sets algorithm for truststore. Corresponds to the truststore option on keytool . |
| Ordinality | Mandatory |
| Valid Values | Defaulted to AES . |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.truststore.alias_keyalg=AES</code> |

com.oracle.ouaf.system.truststore.alias_keysize - Truststore Key Size

| | |
|---------------|--|
| Parameter | <code>com.oracle.ouaf.system.truststore.alias_keysize</code> |
| Description | Sets key size for truststore. Corresponds to the truststore option on keytool . |
| Ordinality | Mandatory |
| Valid Values | Defaulted to 128 . |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>com.oracle.ouaf.system.truststore.alias_keysize=128</code> |

com.oracle.ouaf.system.truststore.file - Location Of Truststore

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.system.truststore.file |
| Description | Location of truststore file. Corresponds to the truststore option on keytool . |
| Ordinality | Mandatory |
| Valid Values | Set to full path location of truststore and name of truststore file. |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.system.truststore.file=/opt/.ouaf_trstore</code> |

com.oracle.ouaf.system.truststore.passwordFileName - Location Of Truststore Password file

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.system.truststore.passwordFileName |
| Description | Location of truststore password file. |
| Ordinality | Mandatory |
| Valid Values | Set to full path location of truststore password file. |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.system.truststore.passwordFileName=/opt/.ouaf_trstorepass</code> |

com.oracle.ouaf.truststore.type - Type Of Truststore

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.truststore.type |
| Description | Type of truststore file. |
| Ordinality | Mandatory |
| Valid Values | Sets Type of Truststore. Defaulted to PKCS12 |
| Source | Value inherited from templates. Use custom template to set this value. (user exit: FW_spl.properties.keystore.truststore.include). |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.system.truststore.type=PKCS12</code> |

com.oracle.ouaf.ui.disableRequiredFieldIndicators – Disable required field indicators on

UI map

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.ui.disableRequiredFieldIndicators |
| Description | Specifies if the required field indicators on the application UI map should be disabled. |
| Ordinality | Optional |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value. |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.ui.disableRequiredFieldIndicators=true</code> |

com.oracle.ouaf.uriSubstitutionVariables.file – URI predefined variables list file

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.uriSubstitutionVariables.file |
| Description | Points to the file containing the predefined URI substitution variable list. |
| Ordinality | Mandatory for cloud installations. Optional for on premise installations. |
| Valid Values | Path of the URI substitution variable list file. |
| Source | Value inherited from <code>CLOUD_SUBSTITUTION_VARIABLE_LIST_FILE_LOCATION</code> in the <code>ENVIRON.INI</code> file. |
| | The following substitution variables are provided by default in OUAF: <ol style="list-style-type: none"> 1. F1_CMA_FILES <i>(The below variables are provided for this release, but will be removed from future releases as these are not used in any product functionality.)</i> 2. F1_BI_EXTRACTS 3. F1_INTERNAL_FILES 4. F1_CUST_APP_BASE 5. F1_PROCESS_DIR 6. F1_SVC_CATALOG_WSDL_DIR 7. F1_PDB_EXTRACTS |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.uriSubstitutionVariables.file=@SPLEBASE@/etc/substitutionVariableList.xml</code> |

com.oracle.ouaf.uriValidation.enable – Enable Restriction of URLs

| | |
|-------------|--|
| Parameter | com.oracle.ouaf.uriValidation.enable |
| Description | Enable URL validation. Restricts external URLs based on white list filename. |

| | |
|---------------|---|
| Ordinality | Mandatory for cloud installations. Optional for on premise installations. |
| Valid Values | [true false]. Default value is false |
| Source | Defaulted from the CLOUD_RESTRICTION_URLS_ENABLE value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.urivalidation.enable=false</code> |

com.oracle.ouaf.whitelist.file – Oracle URL whitelist filename and location

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.whitelist.file |
| Description | Full directory path to the URL oracle whitelist file to be checked for restricting URIs. |
| Ordinality | Mandatory if restriction is activated. |
| Valid Values | Location of whitelist URI file, along with filename. |
| Source | Defaulted from the CLOUD_WHITE_LIST_PATH value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.whitelist.file=@SPLEBASE@/etc/whitelist.xml</code> . The installed product base directory is defaulted from SPLEBASE value in ENVIRON.INI |

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 | Uses XQUERIES_TO_CACHE value in ENVIRON.INI .. |
| Applicability | BAS MOB BATCH |
| Example: | <code>com.oracle.XPath.LRUSize=2000</code> |

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, <i>Value</i> - Time in seconds between flushes. |

| | |
|---------------|--|
| Source | Uses XQUERY_CACHE_FLUSH_TIMEOUT value in ENVIRON.INI . |
| Applicability | BAS BATCH MOB |
| Example: | <code>com.oracle.XPath.flushTimeout=86400</code> |

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: | <code>com.splwg.batch.cluster.jvmName=onlineBatch</code> |

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 <i>Pending</i> 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: | <code>com.splwg.batch.scheduler.daemon=false</code> |

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 | [THIN CLUSTERED] |

| | |
|---------------|--|
| Source | Derived from BATCH_MODE parameter from ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.splwg.grid.executionMode=CLUSTERED</code> |

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 responsible for managing and executing jobs allocated by the Batch Daemon. It is highly recommended to specify a value of false for Production environments. <hr/> <i>Note: The Batch Server is only recommended to be used in Testing, Development or Demonstration environments.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Derived from BATCHENABLED parameter from ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.splwg.grid.online.enabled=false</code> |

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: | <code>com.splwg.grid.distThreadPool.threads.DEFAULT=5</code> |

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 template to set this value to false , if desired. |
| Applicability | WEB BAS |
| Example: | <code>com.sp1wg.schema.newValidations.F1=true</code> |

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. |
| Ordinality | Mandatory |
| Valid Values | Relative location of file from <code>\$\$SPLEBASE/\$\$SPLEBASE%</code> . Defaults to scripts/ouaf.jmx.access.file |
| Source | Value inherited from templates. Use custom template to set this value to the desired location and file, if desired. |
| Applicability | WEB BAS |
| Example: | <code>jmx.remote.x.access.file=scripts/ouaf.jmx.access.file</code> |

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. |
| Ordinality | Mandatory |
| Valid Values | Relative location of file from <code>\$\$SPLEBASE/\$\$SPLEBASE%</code> . Defaults to scripts/ouaf.jmx.password.file |
| Source | Value inherited from templates. Use custom template to set this value to the desired location and file, if desired. |
| Applicability | WEB BAS |
| Example: | <code>jmx.remote.x.password.file=scripts/ouaf.jmx.password.file</code> |

oracle.ouaf.jmx.remote.startPort - JMX Override Port

| | |
|--------------|--|
| Parameter | oracle.ouaf.jmx.remote.startPort |
| Description | Alternative port number for JMX monitoring. Allows for overrides from standard port. Overrides spl.runtime.management.rmi.port . |
| Ordinality | Optional |
| Valid Values | Valid port numbers. Defaults to spl.runtime.management.rmi.port |

| | |
|---------------|--|
| Source | Specif. |
| Applicability | WEB BAS |
| Example: | <code>jmx.remote.x.password.file=scripts/ouaf.jmx.password.file</code> |

ouaf.accessibility.features - Enable Accessibility features

| | |
|---------------|---|
| Parameter | ouaf.accessibility.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 template to set this value to false , if desired. |
| Applicability | WEB |
| Example: | <code>ouaf.accessibility.features=true</code> |

ouaf.attachment.disableSpecialCharCheckOnFileName - Disable Special Characters on file name

| | |
|---------------|--|
| Parameter | ouaf.attachment.disableSpecialCharCheckOnFileName |
| Description | Disables (true) or enables (false) check for special charcters used in file name for attachment objects. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to true , if desired. |
| Applicability | BAS |
| Example: | <code>ouaf.attachment.disableSpecialCharCheckOnFileName=false</code> |

ouaf.batch.onlineLogDir - Online Line Batch Log Directory

| | |
|--------------|--|
| Parameter | ouaf.batch.onlineLogDir |
| Description | Sets the location of output from online batch submission and batch scheduler integration. If this property is set, batch logs will be accessible via the Batch Run Tree (application UI). Otherwise, the batch logs download link in the application will be disabled. |
| Ordinality | Optional |
| Valid Values | Valid location. Defaults to: \$SPLOUTPUT (%SPLOUTPUT% on Windows) |

| | |
|---------------|--|
| Source | Value inherited from BATCH_ONLINE_LOG_DIR from ENVIRON.INI . |
| Applicability | BATCH |
| Example: | <code>ouaf.batch.onlineLogDir=/tmp</code> |

ouaf.batch.onlineLogDir.retain – Retain Online Line Batch Log Files

| | |
|---------------|---|
| Parameter | ouaf.batch.onlineLogDir.retain |
| Description | Sets the total number of log files that are to be stored for each batch code and thread number. The log files for the particular batch code and thread number will be deleted except for the latest number of logs specified. |
| Ordinality | Optional |
| Valid Values | Number. |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>ouaf.batch.onlineLogDir.retain=3</code> The above example will keep the last three logs and delete the rest |

ouaf.database.session.setInstallationTimeZone - Set Time Zone for sessions

| | |
|---------------|---|
| Parameter | ouaf.database.session.setInstallationTimeZone |
| Description | Set this property to true if the Time Zone on the installation record via the online <i>Installation Options - Framework</i> menu item (under <i>Main</i> tab) is different to the default timezone on the database. This will ensure session timezone is corrected. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to true , if desired. |
| Applicability | BAS MOB BATCH |
| Example: | <code>ouaf.database.session.setInstallationTimeZone=false</code> |

ouaf.flush.jms.disabled - Global flush supported

| | |
|--------------|--|
| Parameter | ouaf.flush.jms.disabled |
| Description | Enables (false) or disables (true) global flush feature. |
| Ordinality | Optional |
| Valid Values | [true false] |

| | |
|---------------|---|
| Source | Value inherited from templates. Use custom template to set this value to false , if desired. |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>ouaf.flush.jms.disabled=true</code> |

ouaf.flush.jms.connection - Connection Factory for Global Flush

| | |
|---------------|--|
| Parameter | ouaf.flush.jms.connection |
| Description | JNDI Name for Global JMS connection |
| Ordinality | Optional |
| Valid Values | Valid JNDI for JMS Connection Factory. Default: jms/OUFlushConnectionFactory |
| Source | Value inherited from templates. Use custom template to set this value to an alternative value, if desired. |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>ouaf.flush.jms.connection=jms/OUFlushConnectionFactory</code> |

ouaf.flush.jms.requestTopic - Global Flush Request Topic

| | |
|---------------|--|
| Parameter | ouaf.flush.jms.requestTopic |
| Description | JNDI Name for Global JMS Request Topic |
| Ordinality | Optional |
| Valid Values | Valid JNDI for JMS Request Topic. Default: jms/OUFlushRequestTopic |
| Source | Value inherited from templates. Use custom template to set this value to an alternative value, if desired. |
| Applicability | WEB BAS IWS BATCH MOB |
| Example: | <code>ouaf.flush.jms.requestTopic=jms/OUFlushRequestTopic</code> |

ouaf.flush.jms.responseTopic - Global Flush Response Topic

| | |
|--------------|--|
| Parameter | ouaf.flush.jms.responseTopic |
| Description | JNDI Name for Global JMS Response Topic |
| Ordinality | Optional |
| Valid Values | Valid JNDI for JMS Response Topic. Default: jms/OUFlushResponseTopic |
| Source | Value inherited from templates. Use custom template to set this value to an alternative value, if desired. |

Applicability **WEB** **BAS** **IWS** **BATCH** **MOB**
 Example: `ouaf.flush.jms.responseTopic=jms/OUFlushResponseTopic`

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 template](#) 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 template](#) 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 template](#) to set this value to **disabled**, if desired.
 Applicability **WEB**

y

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 template to set this value to disabled , if desired. |
| Applicability | WEB |
| Example: | <code>ouaf.jmx.com.splwg.base.web.mbeans.FlushBean=enabled</code> |

ouaf.list.buffer.limit - List Iteration limit

| | |
|---------------|--|
| Parameter | ouaf.list.buffer.limit |
| Description | The number of iterations a list is allowed to be read before being truncated. A default of 30 is used, if none or an invalid value is specified for this property. |
| Ordinality | Optional |
| Valid Values | Value between 0 - 256 |
| Source | Value inherited from JEE container. Use custom template to set this value, if alternative desired |
| Applicability | BAS |
| Example: | <code>ouaf.list.buffer.limit=30</code> |

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: <code>com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl</code> <code>org.apache.xalan.xsltc.trax.TransformerFactoryImpl</code> |

| | |
|---------------|---|
| Source | Value inherited from JEE container. Use custom template to set this value, if alternative desired |
| Applicability | BAS BATCH |
| Example: | <code>ouaf.messaging.transformerfactory = org.apache.xmlan.xslt.trax.TransformerFactoryImpl</code> |

ouaf.shortcut.ignore.altKeys - Disable Hot Keys

| | |
|---------------|---|
| Parameter | <code>ouaf.shortcut.ignore.altKeys</code> |
| 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 <code>sp1.properties.root.exit.include</code> user exit to set this value. |
| Applicability | WEB |
| Example: | <code>ouaf.shortcut.ignore.altKeys=C,L</code> |

ouaf.timeout.business_object.default - Set Default Business Object Timeout

| | |
|---------------|--|
| Parameter | <code>ouaf.timeout.business_object.default</code> |
| 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 <code>sp1.properties.service.timeouts.exit.include</code> user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.business_object.default=10</code> |

ouaf.timeout.business_object.<bocode> - Set Specific Business Object Timeout

| | |
|-------------|---|
| Parameter | <code>ouaf.timeout.business_object.<bocode></code> |
| Description | Sets the timeout (in seconds) for Business Object designated by <code><bocode></code> which corresponds to the Business Object Name. This overrides the <code>ouaf.timeout.business_object.default</code> 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_1
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 sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.business_object.CMUser=10</code> |

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 sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.business_service.default=10</code> |

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: |
| | <pre>SELECT bus_svc_cd, descr FROM f1_bus_svc_1 where language_cd = 'ENG';</pre> |
| | 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 <code>spl.properties.service.timeouts.exit.include user exit</code> to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.business_service.F1-EmailService=10</code> |

ouaf.timeout.query.default - Set Default Query Timeout

| | |
|---------------|--|
| Parameter | <code>ouaf.timeout.query.default</code> |
| 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 <code>spl.properties.service.timeouts.exit.include user exit</code> to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.query.default=10</code> |

ouaf.timeout.query.<zonecode> - Set Specific Query Timeout

| | |
|-------------|---|
| Parameter | <code>ouaf.timeout.query.<zonecode></code> |
| Description | Sets the timeout (in seconds) for the online query zone designated by <zonecode> which corresponds to the Query Zone. This overrides the <code>ouaf.timeout.query.default</code> 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 find the ZONE_CD's <zonecode> : <pre>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';</pre> 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 sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.business_service.F1-MSTCFGLS=10</code> |

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 sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.script.default=10</code> |

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. <hr/> <i>Note: This timeout does not apply to plug-in scripts.</i> <hr/> To find valid values for Script, use the Script Maintenance function or use the following SQL: <pre>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';</pre> 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 sp1.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 sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.service.default=10</code> |

ouaf.timeout.service.<service> - Set Specific Service Timeout

| | |
|---------------|--|
| Parameter | ouaf.timeout.service.<service> |
| Description | <p>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.</p> <p>To find valid values for Application Service, use the Application Service Maintenance function or use the following SQL:</p> <pre>SELECT SVC_NAME, DESCR from CI_MD_SVC_L where language_cd = 'ENG';</pre> <p>The LANGUAGE_CD in the above query can be altered to other languages for language packs installed</p> |
| Ordinality | Optional |
| Valid Values | 1 - 2147483647 |
| Source | Use the sp1.properties.service.timeouts.exit.include user exit to set this value as outlined in Online Transaction Timeouts . |
| Applicability | BAS |
| Example: | <code>ouaf.timeout.service.CILTUSEP=10</code> |

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 WEB_WLHOST , 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: | <code>ouaf.web.appViewer.coreAppURL= http://localhost:6500/ouaf/loginPage.jsp#fromAppViewer</code> |

ouaf.web.disablePreload - Disable Screen Preload

| | |
|---------------|--|
| Parameter | ouaf.web.disablePreload |
| Description | Enables or disables 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: | <code>ouaf.web.help.coreAppURL= http://localhost:6500/ouaf/loginPage.jsp</code> |

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 WEB_WLHOST , 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: | <code>ouaf.web.help.coreAppURL=../../ouaf/loginPage.jsp</code> |

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 WEB_WLHOST , 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: | <code>ouaf.web.helpRootURL=/ouaf/help/</code> |

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 JEE 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: | <code>ouaf.web.JspHeader.maxAge=28850</code> |

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: | <code>ouaf.web.preloadAllPages=false</code> |

ouaf.web.useOHW - Use Oracle Help for Web engine

| | |
|---------------|--|
| Parameter | ouaf.web.useOHW |
| Description | Use the Oracle Help for Web engine rather than the legacy SPLHelp engine. This setting is only applicable to Oracle WebLogic customers. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to false , if desired. |
| Applicability | WEB |
| Example: | <code>ouaf.web.useOHW=true</code> |

ouaf.weblogic.enableDiagnostics - Enable ECID tracking

| | |
|---------------|--|
| Parameter | ouaf.weblogic.enableDiagnostics |
| Description | Enables (true) or disable (false) ECID feature . |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to true , if desired. |
| Applicability | WEB |
| Example: | <code>ouaf.weblogic.enableDiagnostics = false</code> |

ouaf.webservices.security.defaultPolicy – IWS customized security policy

| | |
|---------------|--|
| Parameter | ouaf.webservices.security.defaultPolicy |
| Description | Sets the customized policy for the deployed IWS. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Value inherited from templates. Use custom template to set this value to true , if desired. |
| Applicability | WEB |
| Example: | <code>ouaf.weblogic.enableDiagnostics = false</code> |

ouaf.ws.defaultUser - Default User for Web Services

| | |
|---------------|---|
| Parameter | ouaf.ws.defaultUser |
| Description | Default user used for authorization for Inbound Web Services if no credentials are provided on the Web Service call. This is provided for backward compatibility for XML Application Integration customers. It is recommended not to be set at all, as to enforce stronger security for Web Services. |
| Ordinality | Optional |
| Valid Values | Valid authorization user for product. A null or blank value enforces security on each Web Service call. |
| Source | Defaulted from templates. Use custom templates to set this value to an alternative desired value. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.defaultUser=AUSER</code> |

ouaf.ws.deploy.appxml.file - Application XML file for Inbound Web Services

| | |
|---------------|---|
| Parameter | ouaf.ws.deploy.appxml.file |
| Description | Location of application.xml file for Web Services. |
| Ordinality | Man |
| Valid Values | Valid Path and location. |
| Source | Derived from template. This should not be changed. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.deploy.appxml.file=/spl/ouaf/iws/resources/application.xml</code> |

ouaf.ws.deploy.ear - EAR file name for Inbound Web Services

| | |
|--------------|--|
| Parameter | ouaf.ws.deploy.ear |
| Description | Name of EAR file used for Inbound Web Services |
| Ordinality | Optional |
| Valid Values | Valid authorization user for product. A null or blank value enforces security on each Web Service call. |
| Source | Derived from template Use custom templates to set this value to an alternative desired value |

Applicability **IWS**
 Example: `ouaf.ws.deploy.ear=webservices.ear`

ouaf.ws.deploy.contextpath - Context Path for Inbound Web Services

Parameter **ouaf.ws.deploy.contextpath**
 Description Default context path used for Inbound Web Services calls
 Ordinality Optional
 Valid Values Valid authorization user for product. A null or blank value enforces security on each Web Service call.
 Source Derived from template Use [custom templates](#) to set this value to an alternative desired value
 Applicability **IWS**
 Example: `ouaf.ws.deploy.contextpath=/ouaf/webservices/`

ouaf.ws.pageService.annotations - Security Annotation for XAI Legacy services

Parameter **ouaf.ws.pageService.annotations**
 Description The default security WS-Policy annotations attached for legacy page services (XAI) within IWS. To ensure backward compatibility, for customers using XAI legacy services, it is recommended to set this value to the default **F1-USERNAM** annotation.
 Ordinality Optional
 Valid Values Valid Web Service Annotation
 Source Defaulted from templates. Use [custom templates](#) to set this value to an alternative desired value.
 Applicability **IWS**
 Example: `ouaf.ws.pageService.annotations=F1-USERNAM`

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: | <code>ouaf.ws.superusers=SYSUSER</code> |

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 <code>\$SPLEBASE/splapp/iws/iws-build.xml</code> (or <code>%%\$SPLEBASE%\splapp\iws\iws-build.xml</code> on Windows) |
| Source | Derived from SPLDIR parameter from ENVIRON.INI . Use custom templates to set this value to an alternative desired value. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.tools.artifact.generated.buildfile=/spl/OUAFDEMO/splapp/iws/iws-build.xml</code> |

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 <code>\$SPLEBASE/splapp/iws/gen</code> (or <code>%%\$SPLEBASE%\splapp\iws\gen</code> on Windows) |
| Source | Derived from SPLDIR parameter from ENVIRON.INI . Use custom templates to set this value to an alternative desired value. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.tools.artifact.generated.sourcedir=/spl/OUAFDEMO/splapp/iws/gen</code> |

ouaf.ws.tools.artifact.resource.dir - Location of 310resources

| | |
|-------------|---|
| 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 SPLDIR parameter from ENVIRON.INI . Use custom templates to set this value to an alternative desired value. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.tools.artifact.resource.dir=/spl/OUAFDEMO/splapp/iws/resources</code> |

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 SPLDIR parameter from ENVIRON.INI . Use custom templates to set this value to an alternative desired value. |
| Applicability | IWS |
| Example: | <code>ouaf.ws.tools.artifact.schema.dir=/spl/OUAFDEMO/splapp/xai/schemas</code> |

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 SPLDIR 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: | <code>ouaf.ws.tracing=true</code> |

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: | <code>ouaf.ws.xslDirOverride=/temp</code> <code>ouaf.ws.xslDirOverride=file:///c:/temp</code> |

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: | <code>ouaf.xai.HTTPBasicAuthPasswordEnc=ENC(unj0oM7+zufXI5r6Q...)</code> |

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_SPLUSER parameter from ENVIRON.INI . |
| Applicability | WEB |
| Example: | <code>ouaf.xai.HTTPBasicAuthUser=SYSUSER</code> |

spl.csv.delimiter.useFromDisplayProfile - List CSV delimiter

| | |
|---------------|--|
| 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: | <code>spl.csv.delimiter.useFromDisplayProfile=false</code> |

spl.ejbContainer.contextFactory - Context Factory for Business Application Server

| | |
|---------------|--|
| Parameter | spl.ejbContainer.contextFactory |
| Description | Default JEE Context Factory to connect tiers. |
| Ordinality | Mandatory |
| Valid Values | Valid connection factory: <u>Oracle WebLogic:</u> weblogic.jndi.WLInitialContextFactory |
| Source | Derived from template. Use custom templates to set this value to an alternative, if desired. |
| Applicability | WEB IWS |
| Example: | <code>spl.ejbContainer.contextFactory=weblogic.jndi.WLInitialContextFactory</code> |

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 encrypted Administration user |
| Source | <u>Oracle WebLogic:</u> WEB_WLSYSPASS from ENVIRON.INI Use custom templates to set this value to an alternative, if desired. |
| Applicability | WEB IWS |
| Example: | <code>spl.ejbContainer.password=ENC(Hi7/Rfy...)</code> |

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: <u>Oracle WebLogic:</u> t3://<server>:<port> or t3s://<server>:<port> |
| Source | Derived from template. Use custom templates to set this value to an alternative, if desired. |
| Applicability | WEB IWS |
| Example: | <code>spl.ejbContainer.url = t3://localhost:6500</code> |

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 JEE Server |
| Source | <u>Oracle WebLogic:</u> WEB_WLSYSUSER 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 JEE 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
 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 Encrypted 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 JEE Context Factory to connect to Appointment Booking Reservation (ABR) component.

Ordinality Mandatory

Valid Values Valid connection factory:

Oracle WebLogic:

weblogic.jndi.WLInitialContextFactory

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 Encrypted 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: | <code>spl.mwm.abr.timeout=120000</code> |

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: <u>Oracle WebLogic:</u> t3://<server>:<port> or t3s://<server>:<port> |
| Source | Derived from URL from ENVIRON.INI |
| Applicability | MOB |
| Example: | <code>spl.abr.url = t3://localhost:7751</code> |

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 connection user for GIS. |
| Source | Derived from WLS_USERID from ENVIRON.INI |
| Applicability | MOB |
| Example: | <code>spl.mwm.abr.userid=ENC(ho4tqPhP...)</code> |

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: | <code>spl.mwm.scheduler.abr.maxProcessingTime=5</code> |

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: | <code>spl.mwm.scheduler.abr.minRequests=1</code> |

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: | <code>spl.mwm.scheduler.cleanse.interval=900</code> |

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: | <code>spl.mwm.scheduler.connectionTimeout=300000</code> |

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: | <code>spl.mwm.scheduler.maintenanceCycleTime=300</code> |

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: | <code>spl.mwm.scheduler.mapDir=/spl/splapp/OUAFDEMO/sploutput</code> |

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: | <code>spl.mwm.scheduler.nodeId=MyNode</code> |

spl.runtime.mwm.scheduler.ipcStartPort - IPC Start Port

| | |
|---------------|--|
| Parameter | <code>spl.runtime.mwm.scheduler.ipcStartPort</code> |
| Description | Starting port number for Inter Process Communication (IPC) between the scheduler and products. |
| Ordinality | Mandatory |
| Valid Values | Valid port numbers. Schedulers allocate port numbers starting from this number. |
| Source | Derived from <code>IPCSTARTPORT</code> from ENVIRON.INI |
| Applicability | MOB |
| Example: | <code>spl.runtime.mwm.scheduler.ipcStartPort=7568</code> |

spl.runtime.compatibility.uiMapDisableInputValue - Map Defaults

| | |
|---------------|--|
| Parameter | <code>spl.runtime.compatibility.uiMapDisableInputValue</code> |
| Description | <p>Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.1 in relation to blanking out default values in screens.</p> <p>Setting this value to true will emulate Oracle Utilities Application Framework V2.1 behavior.</p> <p>This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.1 and wish to retain that behavior.</p> |
| Ordinality | Optional |
| Valid Values | [true false] |
| Source | Manually set. Use user exit <code>spl.properties.exit.include</code> to set this value to true , if desired |
| Applicability | WEB |
| Example: | <code>spl.runtime.compatibility.uiMapDisableInputValue=false</code> |

spl.runtime.compatibility.uiMapDropdownSelectFirstValue - Dropdown defaults

| | |
|-------------|---|
| Parameter | <code>spl.runtime.compatibility.uiMapDropdownSelectFirstValue</code> |
| Description | <p>Enables or disables compatibility behavior with Oracle Utilities Application Framework V2.1 in relation to defaulting to the first value in dropdownlist.</p> <p>Setting this value to true will emulate Oracle Utilities Application Framework V2.1 behavior.</p> <p>This setting is only applicable to customers who are upgrading from Oracle Utilities Application Framework V2.1 and wish to retain that behavior.</p> |
| 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: | <code>spl.runtime.compatibility.uiMapDropdownSelectFirstValue=false</code> |

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: | <code>spl.runtime.compatibility.uiMapDisableTitle=false</code> |

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 identifier'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.customSQLSecurity – Enable SQL Whitelist

Parameter **spl.runtime.customSQLSecurity**

Description Enables or disables SQL whitelist validations for CM data explorer zones. Setting this to true will restrict the SQL in CM data explorer zone to the whitelist. Setting this to false will not restrict the SQL in CM data explorer zones.

Ordinality Mandatory

Valid Values [true|false]

Source Manually set.

Applicability **WEB**

Example: `spl.runtime.customSQLSecurity=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 <code>spl.runtime.environ.SPLEBASE</code> with the <code>etc</code> directory appended. |
| Source | Derived from cistab file. |
| Applicability | WEB BAS IWS |
| Example: | <code>spl.runtime.environ.init.dir=/spl/OUAFDEMO/etc</code> |

`spl.runtime.environ.isWebExpanded` - Expanded Mode

| | |
|---------------|---|
| Parameter | <code>spl.runtime.environ.iswebExpanded</code> |
| 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 <code>WEB_IEXPANDED</code> parameter from ENVIRON.INI . |
| Applicability | WEB BAS |
| Example: | <code>spl.runtime.environ.iswebExpanded=false</code> |

`spl.runtime.environ.SPLEBASE` - Location Of Software

| | |
|---------------|---|
| Parameter | <code>spl.runtime.environ.SPLEBASE</code> |
| 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 BATCH |
| Example: | <code>spl.runtime.environ.SPLEBASE=/spl/OUAFDEMO</code> |

`spl.runtime.initialize.waittime` - Start Time Delay

| | |
|-------------|---|
| Parameter | <code>spl.runtime.initialize.waittime</code> |
| Description | Delay Time (in seconds) for start of application from initiation time. This is used to delay startup to ensure downstream connections are established prior to startup. |
| Ordinality | Optional |

| | |
|---------------|--|
| Valid Values | 0 - 3000 (Default: 45) |
| Source | Derived from templates. Use custom templates to set this value to an alternative, if desired |
| Applicability | IWS |
| Example: | <code>spl.runtime.initialize.waittime = 45</code> |

spl.runtime.envIRON.SPLOUTPUT - Location Of Output

| | |
|---------------|--|
| Parameter | <code>spl.runtime.envIRON.SPLOUTPUT</code> |
| 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 BATCH |
| Example: | <code>spl.runtime.envIRON.SPLEBASE=/spl/splapp/OUAFDEMO</code> |

spl.runtime.management.batch.cluster.url - Batch Cluster URL

| | |
|---------------|--|
| Parameter | <code>spl.runtime.management.batch.cluster.url</code> |
| Description | URL for the Batch Cluster |
| Ordinality | Mandatory |
| Valid Values | Valid URL for Batch Cluster |
| Source | Derived from <code>WEB_BATCH_CLUSTER_URL</code> parameter from ENVIRON.INI . |
| Applicability | WEB |
| Example: | <code>spl.runtime.management.batch.cluster.url=</code> |

spl.runtime.management.connector.url.default - JMX URL

| | |
|--------------|--|
| Parameter | <code>spl.runtime.management.connector.url.default</code> |
| 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 <code>WEB_WLHOST</code> and <code>WEB_JMX_RMI_PORT_PERFORMANCE</code> Business Application Server <code>BSN_WLHOST</code> and <code>BSN_JMX_RMI_PORT_PERFORMANCE</code> |

Inbound Web Services**BSN_WLHOST** and **BSN_JMX_RMI_PORT_PERFORMANCE**

| | |
|---------------|--|
| Source | Derived from ENVIRON.INI file. |
| Applicability | WEB BAS IWS |
| Example: | <code>spl.runtime.environ.SPLEBASE= service:jmx:rmi:///jndi/rmi://localhost: 1100/oracle/ouaf/ejbAppConnector</code> |

spl.runtime.management.rmi.port - JMX Port

| | |
|---------------|---|
| Parameter | spl.runtime.management.rmi.port |
| Description | Port number to be used for JMX based monitoring . |
| Ordinality | Optional |
| Valid Values | Port Number is generated from the following settings: <u>Web Application Server</u> WEB_JMX_RMI_PORT_PERFORMANCE <u>Business Application Server</u> BSN_JMX_RMI_PORT_PERFORMANCE <u>Inbound Web Services</u> BSN_JMX_RMI_PORT_PERFORMANCE <u>Batch</u> BATCH_RMI_PORT |
| Source | Derived from ENVIRON.INI file. |
| Applicability | WEB BAS IWS |
| Example: | <code>spl.runtime.management.rmi.port=1100</code> |

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.onserver=nodes parameter. |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Derived from ONSCONFIG parameter from ENVIRON.INI . |
| Applicability | BAS BATCH |
| Example: | <code>spl.runtime.options.isFCFEnabled=false</code> |

spl.runtime.options.onserver=nodes - ONS Configuration

| | |
|---------------|---|
| Parameter | spl.runtime.options.onserver=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: | <code>spl.runtime.options.onserver=nodes=racnode1:4200,racnode2:4200\nwalletfile= /oracle11/onswalletfile</code> |

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: | <code>spl.runtime.options.allowSystemDateOverride=false</code> |

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 reenabled manually.
- Screen Preloading is disabled. This can be reenabled manually.
- Compression is disabled at all layers of the architecture. This can be reenabled manually.
- The product is set to expanded mode to allow developers access to individual JEE 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**

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 exit 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.sql.highValue - SQL High Value

| | |
|--------------|--|
| Parameter | spl.runtime.sql.highValue |
| Description | The high value used for parameters 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** (CHAR_BASED_DB) setting in the [ENVIRON.INI](#)
 Applicability **WEB** **BAS** **IWS** **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**
 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** **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 BATCH |
| Example: | <code>hibernate.cache.use_second_level_cache = false</code> |

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 |
| Example: | <code>hibernate.connection.datasource=jdbc/ouafdb</code> |

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: <code>oracle.jdbc.driver.OracleDriver</code> |
| Source | Maintained by product templates. Use custom templates to implement a different value. |

| | |
|---------------|--|
| Applicability | BAS IWS BATCH |
| Example: | <code>hibernate.connection.driver_class = oracle.jdbc.driver.OracleDriver</code> |

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 BATCH |
| Example: | <code>hibernate.connection.password = ENCKS(...)</code> |

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 JDBC 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 BATCH |
| Example: | <code>hibernate.connection.provider_class=com.splwg.shared.common.UCPConnectionProvider</code> |

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 BATCH |
| Example: | <code>hibernate.connection.release_mode=after_transaction</code> |

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 BATCH |
| Example: | <code>hibernate.connection.url = jdbc:oracle:thin:@ouafdemo.us.oracle.com:1522:/MYPDB</code> |

hibernate.connection.username - Database User

| | |
|---------------|---|
| Parameter | hibernate.connection.username |
| Description | JDBC User for connection. |
| Ordinality | Mandatory |
| Valid Values | The value depends on the channel accessed: 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 BATCH |
| Example: | <code>hibernate.connection.username = CISUSER</code> |

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 BATCH |
| Example: | <code>hibernate.dialect = org.hibernate.dialect.Oracle10gDialect</code> |

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 BATCH |
| Example: | <code>hibernate.jdbc.batch_size = 30</code> |

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 BATCH |
| Example: | <code>hibernate.jdbc.fetch_size = 100</code> |

hibernate.max_fetch_depth - Join Fetch Depth

| | |
|-------------|---|
| Parameter | hibernate.max_fetch_depth |
| Description | Sets a maximum <i>depth</i> 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 BATCH |
| Example: | <code>hibernate.max_fetch_depth = 2</code> |

hibernate.query.factory_class - HQL Parser

| | |
|---------------|--|
| Parameter | hibernate.query.factory_class |
| Description | The classname of a Query 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 BATCH |
| Example: | <code>hibernate.query.factory_class= org.hibernate.hql.internal.classic.ClassicQueryTranslatorFactory</code> |

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 BATCH |
| Example: | <code>hibernate.query.substitutions = true 'Y', false 'N'</code> |

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 BATCH |
| Example: | <code>hibernate.show_sql = false</code> |

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 BATCH |
| Example: | <code>hibernate.transaction.factory_class = org.hibernate.transaction.JDBCTransactionFactory</code> |

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 BATCH |
| Example: | <code>hibernate.ucp.connection_wait_timeout=5</code> |

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** **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** **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** **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** **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 BATCH |
| Example: | <code>hibernate.ucp.min_size=1</code> |

submitbatch.properties

The submitbatch.properties controls the behavior of individual batch job threads within the batch architecture. This configuration file exists in two forms:

- A global **submitbatch.properties** that defines global defaults for the implementation.
- A set of **submitbatch.properties** files for individual jobs that defines any configuration overrides or specific configuration settings for an individual job.

This section outlines each setting for the configuration file.

Note: It is recommended to use the [bedit](#) configuration utility to maintain this configuration file.

com.oracle.ouaf.system.keystore.alias - KeyStore Alias

| | |
|---------------|---|
| Parameter | com.oracle.ouaf.system.keystore.alias |
| Description | Alias used within keystore. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_ALIAS parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.alias=thealias</code> |

com.oracle.ouaf.system.keystore.file - KeyStore File

| | |
|-------------|---|
| Parameter | com.oracle.ouaf.system.keystore.file |
| Description | Location and name of keystore file for encryption keys. |

| | |
|---------------|---|
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_KEYSTORE_FILE parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.file =/scratch/OUAF/ks/.ouaf_keystore</code> |

com.oracle.ouaf.system.keystore.hmac_key_alias - KeyStore Mode

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.hmac_key_alias</code> |
| Description | HMAC Alias used within keystore |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_HMAC_ALIAS parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.hmac_key_alias=thealiasHMAC</code> |

com.oracle.ouaf.system.keystore.mode - KeyStore Mode

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.mode</code> |
| Description | Padding mode used within keystore |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_MODE parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.mode=CBC</code> |

com.oracle.ouaf.system.keystore.padding - KeyStore Padding Algorithm

| | |
|--------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.padding</code> |
| Description | Padding algorithm used within keystore |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_PADDING parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |

Applicability **BATCH**
 Example: `com.oracle.ouaf.system.keystore.padding=PKCS5Padding`

com.oracle.ouaf.system.keystore.passwordFileName - KeyStore File

Parameter **com.oracle.ouaf.system.keystore.passwordFileName**
 Description Location and name of keystore password file.
 Ordinality Mandatory
 Valid Values Generated using the [KS_STOREPASS_FILE](#) parameter.
 Source Maintained by product templates. Use [custom templates](#) to implement a different value.
 Applicability **BATCH**
 Example: `com.oracle.ouaf.system.keystore.passwordFileName=/scratch/OUAF/ks/.ouaf_storepass`

com.oracle.ouaf.system.keystore.type - KeyStore Type

Parameter **com.oracle.ouaf.system.keystore.type**
 Description Type of keystore.
 Ordinality Mandatory
 Valid Values Generated using the [KS_STORETYPE](#) parameter.
 Source Maintained by product templates. Use [custom templates](#) to implement a different value.
 Applicability **BATCH**
 Example: `com.oracle.ouaf.system.keystore.type=JCEKS`

com.ouaf.batch.jvmoptions - JVM Options

Parameter **com.ouaf.batch.jvmoptions**
 Description Custom JVM Options for submitter.

*Note: This setting can be overridden using the **-z** option on the [submitjob\[.sh\]](#) command line.*

 Ordinality Mandatory
 Valid Values Valid JVM Options for the Java version used
 Source Maintained by product templates. Use [bedit](#) to implement a different value.
 Applicability **BATCH**
 Example: `com.ouaf.batch.jvmoptions=-Xms1024k -Xmx2048k -XX:MaxPermSize=512m`

com.splwg.grid.executionMode - Mode of Execution

| | |
|---------------|---|
| Parameter | com.splwg.grid.executionMode |
| Description | Execution Mode used by process. <i>Note: This setting can be overridden using the -e option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | [CLUSTERED THIN] <i>Note: The THIN mode is used by the Oracle Utilities SDK and should not be used outside that tool.</i> |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.grid.executionMode=CLUSTERED</code> |

com.splwg.batch.submitter - THIN Submitter

| | |
|---------------|--|
| Parameter | com.splwg.batch.submitter |
| Description | Whether this execution can run without a threadpoolworker. Reserved for development of batch using the Oracle Utilities SDK. If set to true then, com.splwg.grid.executionMode must be set to THIN . |
| Ordinality | Mandatory |
| Valid Values | [false true] |
| Source | Maintained by product templates for LOCAL submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter=true</code> |

com.splwg.batch.submitter.batchCd - Batch Code

| | |
|---------------|--|
| Parameter | com.splwg.batch.submitter.batchCd |
| Description | Batch Code for bbatch specific configuration. This setting should not be set in the global submitbatch.properties file. <i>Note: This setting should only be set in configuration files for testing purposes. In other situations the -b option on the submitjob[.sh] command line should be used.</i> |
| Ordinality | Mandatory |
| Valid Values | Batch Code |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |

Example: `com.splwg.batch.submitter.batchCd=QABATCH`

com.splwg.batch.submitter.distThreadPool - Threadpool

| | |
|---------------|--|
| Parameter | <code>com.splwg.batch.submitter.distThreadPool</code> |
| Description | Name of pool to be used for batch process. If threadpoolworker not used then LOCAL must be specified. This can be set globally and at the individual batch job level. <i>Note: This setting can be overridden using the -p option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Default: DEFAULT |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.distThreadPool=DEFAULT</code> |

com.splwg.batch.submitter.languageCd - Language Code

| | |
|---------------|--|
| Parameter | <code>com.splwg.batch.submitter.languageCd</code> |
| Description | Language code used for messages for execution of background process. Relevant language pack must be installed. <i>Note: This setting can be overridden using the -l option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Default: ENG |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.languageCd=ENG</code> |

com.splwg.batch.submitter.maximumCommitRecords - Commit Interval

| | |
|-------------|--|
| Parameter | <code>com.splwg.batch.submitter.maximumCommitRecords</code> |
| Description | Commit Interval used for execution. This is the number of objects processed per work unit. <i>Note: This setting can be overridden using the -f option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |

| | |
|---------------|---|
| Valid Values | Job specific |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.maximumCommitRecords=200</code> |

com.splwg.batch.submitter.promptForValues - Prompt for Values

| | |
|---------------|--|
| Parameter | com.splwg.batch.submitter.promptForValues |
| Description | Prompt for value when running in interactive mode. This is primarily used for developers who are testing their job manually. <hr/> <i>Note: Implementers should only use the true value if executing jobs from the Oracle Utilities SDK.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.promptForValues=false</code> |

com.splwg.batch.submitter.processDate - Business Date

| | |
|---------------|---|
| Parameter | com.splwg.batch.submitter.processDate |
| Description | Business Date for batch process. This setting should not be set in the global submitbatch.properties file. <hr/> <i>Note: This setting should only be set in configuration files for testing purposes. In other situations the -d option on the submitjob[.sh] command line should be used.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | Business Date in ISO YYYY-MM-DD format |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.processDate=2015-03-08</code> |

com.splwg.batch.submitter.rerunNumber - Re Run Number

| | |
|-------------|---|
| Parameter | com.splwg.batch.submitter.rerunNumber |
| Description | Rerun Number to be used for execution. <hr/> <i>Note: Only batch controls that support rerun numbers are supported with this</i> |

configuration setting.

*Note: This setting can be overridden using the **-r** option on the [submitjob\[.sh\]](#) command line.*

| | |
|---------------|---|
| Ordinality | Optional |
| Valid Values | Valid run number. A value of zero (0) denotes disables this setting. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.rerunNumber=56</code> |

com.splwg.batch.submitter.softParameter - Job Parameters

| | |
|---------------|--|
| Parameter | com.splwg.batch.submitter.softParameter |
| Description | For any program-specific parameters, use this form of property specification in the format: <code>com.splwg.batch.submitter.softParameter.<parmname>=<value></code> The <parmname> denotes the name of the parameter. For example, to specify a <i>number of rows to skip</i> when submitting a validation program: <code>com.splwg.batch.submitter.softParameter.SKIP-ROWS=1000</code> Multiple soft parameters may be specified. |
| | <i>Note: This setting can be overridden using the -x or -X options on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Valid value for parameter |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.softParameter.SKIP-ROWS=1000</code> |

com.splwg.batch.submitter.threadCount - Thread Limit

| | |
|--------------|--|
| Parameter | com.splwg.batch.submitter.threadCount |
| Description | Thread limit used for execution. <i>Note: Only batch controls that support multithreaded are supported with this configuration setting.</i> <i>Note: This setting can be overridden using the -c option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Valid thread limit. A value of one (1) is used for single threaded jobs. Default: 1 |

| | |
|---------------|---|
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.threadCount=1</code> |

com.splwg.batch.submitter.threadNumber - Thread Number

| | |
|---------------|--|
| Parameter | com.splwg.batch.submitter.threadNumber |
| Description | Individual thread number used for execution. <hr/> <i>Note: This setting can be overridden using the -t option on the submitjob[.sh] command line.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | Valid thread number. The value must be less than or equal to the com.splwg.batch.submitter.threadCount . A value of zero (0) spawns multiple threads up to the limit expressed in com.splwg.batch.submitter.threadCount . Default: 0 |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.threadNumber=0</code> |

com.splwg.batch.submitter.traceProgramEnd - Trace Program End

| | |
|---------------|---|
| Parameter | com.splwg.batch.submitter.traceProgramEnd |
| Description | Enables or disables tracing for the end of the program. <hr/> <i>Note: This setting is designed for developers and for support purposes.</i> <i>Note: This setting can be overridden using the -g option on the submitjob[.sh] command line.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.traceProgramEnd=false</code> |

com.splwg.batch.submitter.traceProgramStart - Trace Program Start

| | |
|-------------|---|
| Parameter | com.splwg.batch.submitter.traceProgramStart |
| Description | Enables or disables tracing for the start of the program. |

Note: This setting is designed for developers and for support purposes.

*Note: This setting can be overridden using the **-g** option on the [submitjob\[.sh\]](#) command line.*

| | |
|---------------|--|
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.traceProgramStart=false</code> |

com.splwg.batch.submitter.traceSQL - Trace SQL Statements

| | |
|---------------|--|
| Parameter | <code>com.splwg.batch.submitter.traceSQL</code> |
| Description | Enables or disables tracing of ALL SQL statements executed by the program. <i>Note: This setting is designed for developers and for support purposes.</i> <i>Note: This setting can be overridden using the -g option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.traceSQL=false</code> |

com.splwg.batch.submitter.traceStandardOut - Trace Output Statements

| | |
|---------------|---|
| Parameter | <code>com.splwg.batch.submitter.traceStandardOut</code> |
| Description | Enables or disables tracing of ALL in-program debug statements executed by the program. <i>Note: This setting is designed for developers and for support purposes.</i> <i>Note: This setting can be overridden using the -g option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.traceStandardOut=false</code> |

com.splwg.batch.submitter.userId - Security user

| | |
|---------------|---|
| Parameter | com.splwg.batch.submitter.userId |
| Description | Userid used by job to check authorization to underlying application services. <i>Note: This setting can be overridden using the -u option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Valid userid. <i>Note: The SYSUSER account should not be used for non-demonstration environments.</i> |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.submitter.userId=USER01</code> |

spl.runtime.batch.L2CacheMode - L2 Cache control

| | |
|---------------|--|
| Parameter | spl.runtime.batch.L2CacheMode |
| Description | Defines the 2nd-level cache mode. Valid values are READ_ONLY (default), READ_WRITE or OFF . <ul style="list-style-type: none"> • READ_ONLY is for performance as it loads the configuration entities into the 2nd-level (JVM-level) cache and accesses them from there only. In READ_ONLY mode, the cached entities may not be updated. • READ_WRITE also loads them into the L2 cache, but they can be updated. The drawback is that an update of a cached entity hits both the database and cache as it gets updated in both places. • In OFF mode the 2nd-level cache is completely disabled. Some jobs, for example LDAP import (F1-LDAP), need to update some of these entities (e.g. User), so it requires the L2 cache to be either OFF or READ_WRITE. The choice between OFF and READ_WRITE depends on requirement; the effect is the same. <i>Note: This setting can be overridden using the -l2 option on the submitjob[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | [READ_ONLY READ_WRITE OFF] |
| Source | Maintained by product templates for submission. |
| Applicability | BATCH |
| Example: | <code>spl.runtime.batch.L2CacheMode=READ_ONLY</code> |

tangosol.coherence.distributed.localstorage - Cache Mode

| | |
|---------------|---|
| Parameter | tangosol.coherence.distributed.localstorage |
| Description | Sets whether data is cached local or not. This is used to optimize large networks of submitters and threadpools with cache nodes. |
| Ordinality | Mandatory |
| Valid Values | [false true] |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.distributed.localstorage=false</code> |

tangosol.coherence.role - Submitter Role

| | |
|---------------|---|
| Parameter | tangosol.coherence.role |
| Description | Sets role of Submitter. Informational information on JVM used for monitoring purposes only. |
| Ordinality | Optional |
| Valid Values | Generated with the name OAAF_<batchcode>_submitter where <batchcode> is the Batch Control identifier for the batch job. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.role=OAAF_QABATCH_Submitter</code> |

threadpoolworker.properties

This configuration sets the threadpoolworker configuration for global and individual threadpoolworkers. As with submitters there are global defaults as well as individual threadpoolworker configurations.

Note: Some settings in this file are shared with the [submitbatch.properties](#) file but these settings apply to the threadpoolworker JVM's rather than submitter JVM's.

Note: This section is the default set of configuration parameters. Additional [Oracle Coherence parameters](#) can be added using [custom templates](#).

com.oracle.ouaf.batch.scriptDir - base scripts directory for infrastructure job

| | |
|-------------|---|
| Parameter | com.oracle.ouaf.batch.scriptDir |
| Description | The base directory in which the scripts have to be placed for the generic infrastructure job to execute them. |

| | |
|---------------|--|
| Ordinality | Optional |
| Valid Values | Directory path. |
| Source | Defaulted by configureEnv. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.batch.scriptDir=\$SPLEBASE/scripts/generic/batch/exposed/\$OWNER</code> <p>\$OWNER would be the owner flag of the batch control thereby allowing restrictions on which scripts might be executed. Additional owners can be provided access by creating symbolic links in those owners' directories.</p> |

com.oracle.ouaf.system.keystore.alias - Threadpoolworker KeyStore Alias

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.alias</code> |
| Description | Alias used within keystore for threadpoolworker. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_ALIAS parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.alias=thealias</code> |

com.oracle.ouaf.system.keystore.file - Threadpoolworker KeyStore File

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.file</code> |
| Description | Location and name of keystore file for encryption keys for threadpoolworker. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_KEYSTORE_FILE parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.file=/scratch/OUAF/ks/.ouaf_keystore</code> |

com.oracle.ouaf.system.keystore.hmac_key_alias - Threadpoolworker KeyStore Mode

| | |
|-------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.hmac_key_alias</code> |
| Description | HMAC Alias used within keystore for threadpoolworker. |
| Ordinality | Mandatory |

| | |
|---------------|---|
| Valid Values | Generated using the KS_HMAC_ALIAS parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.hmac_key_alias=thealiasHMAC</code> |

com.oracle.ouaf.system.keystore.mode - Threadpoolworker KeyStore Mode

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.mode</code> |
| Description | Padding mode used within keystore for threadpoolworker. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_MODE parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.mode=CBC</code> |

com.oracle.ouaf.system.keystore.padding - Threadpoolworker KeyStore Padding Algorithm

| | |
|---------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.padding</code> |
| Description | Padding algorithm used within keystore for threadpoolworker. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_PADDING parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.oracle.ouaf.system.keystore.padding=PKCS5Padding</code> |

com.oracle.ouaf.system.keystore.passwordFileName - Threadpoolworker KeyStore File

| | |
|--------------|---|
| Parameter | <code>com.oracle.ouaf.system.keystore.passwordFileName</code> |
| Description | Location and name of keystore password file for threadpoolworker. |
| Ordinality | Mandatory |
| Valid Values | Generated using the KS_STOREPASS_FILE parameter. |
| Source | Maintained by product templates. Use custom templates to implement a different value. |

Applicability **BATCH**

Example: `com.oracle.ouaf.system.keystore.passwordFileName=/scratch/OUAF/ks/.ouaf_storepass`

com.oracle.ouaf.system.keystore.type - Threadpoolworker KeyStore Type

Parameter `com.oracle.ouaf.system.keystore.type`

Description Type of keystore for threadpoolworker.

Ordinality Mandatory

Valid Values Generated using the [KS_STORETYPE](#) parameter.

Source Maintained by product templates. Use [custom templates](#) to implement a different value.

Applicability **BATCH**

Example: `com.oracle.ouaf.system.keystore.type=JCEKS`

com.oracle.ouaf.system.truststore.file - Location Of Truststore

Parameter `com.oracle.ouaf.system.truststore.file`

Description Location of truststore file. Corresponds to the **truststore** option on **keytool**.

Ordinality Mandatory

Valid Values Set to full path location of truststore and name of truststore file.

Source Defaulted from the [TS_KEYSTORE_FILE](#) value in [ENVIRON.INI](#).

Applicability **BAS**

Example: `com.oracle.ouaf.system.truststore.file=/opt/.ouaf_trstore`

com.oracle.ouaf.system.truststore.passwordFileName - Location Of Truststore Password file

Parameter `com.oracle.ouaf.system.truststore.passwordFileName`

Description Location of truststore password file.

Ordinality Mandatory

Valid Values Set to full path location of truststore password file.

Source Defaulted from the [TS_STOREPASS_FILE](#) value in [ENVIRON.INI](#).

Applicability **BAS**

Example: `com.oracle.ouaf.system.truststore.passwordFileName=/opt/.ouaf_trstorepass`

com.oracle.ouaf.truststore.type - Type Of Truststore

| | |
|---------------|--|
| Parameter | com.oracle.ouaf.truststore.type |
| Description | Type of truststore file. |
| Ordinality | Mandatory |
| Valid Values | Sets Type of Truststore. |
| Source | Defaulted from the TS_STORETYPE value in ENVIRON.INI . |
| Applicability | BAS |
| Example: | <code>com.oracle.ouaf.system.truststore.type=JCEKS</code> |

com.ouaf.batch.disableAutoFlush – Auto flush batch L2 cache

| | |
|---------------|---|
| Parameter | com.ouaf.batch.disableAutoFlush |
| Description | Disables the auto flushing of batch L2 cache. If this value is set to true, the L2 cache auto flush will be disabled. |
| Ordinality | Mandatory |
| Valid Values | [true false]. Default value is false. |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BAS |
| Example: | <code>com.ouaf.batch.disableAutoFlush=true</code> |

com.ouaf.batch.flushIntervalInSeconds – Batch L2 cache auto flush interval

| | |
|---------------|--|
| Parameter | com.ouaf.batch.flushIntervalInSeconds |
| Description | Sets the time interval between batch L2 cache auto flushes. The time interval should be specified in seconds. Default value is 60. |
| Ordinality | Optional |
| Valid Values | Value in seconds. Default is 60. |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BAS |
| Example: | <code>com.ouaf.batch.flushIntervalInSeconds=90</code> |

com.ouaf.batch.jvmoptions - Threadpool JVM Options

| | |
|-------------|------------------------------------|
| Parameter | com.ouaf.batch.jvmoptions |
| Description | Custom JVM Options for threadpool. |

*Note: This setting can be overridden using the **-z** option on the [threadpoolworker\[.sh\]](#) command line.*

| | |
|---------------|--|
| Ordinality | Mandatory |
| Valid Values | Valid JVM Options for the Java version used |
| Source | Maintained by product templates. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.ouaf.batch.jvmoptions=-Xms1024k -Xmx2048k -XX:MaxPermSize=512m</code> |

com.splwg.grid.executionMode - Threadpool Mode of Execution

| | |
|---------------|--|
| Parameter | <code>com.splwg.grid.executionMode</code> |
| Description | Execution Mode used by this threadpool. <i>Note: This setting can be overridden using the -e option on the threadpoolworker[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | [CLUSTERED] |
| Source | Maintained by product templates. |
| Applicability | BATCH |
| Example: | <code>com.splwg.grid.executionMode=CLUSTERED</code> |

com.splwg.grid.distThreadPool.threads - Threadpool Thread Limit

| | |
|---------------|--|
| Parameter | <code>com.splwg.grid.distThreadPool.threads</code> |
| Description | Maximum number of threads for a particular threadpool. The format of the setting is: <code>com.splwg.grid.distThreadPool.threads.<poolname>=<limit></code> where: <code><poolname></code> is the Name of the threadpool <code><limit></code> is the thread limit supported by each instance of the threadpool |
| | <i>Note: This setting can be overridden using the -p option on the threadpoolworker[.sh] command line.</i> |
| Ordinality | Mandatory |
| Valid Values | Threadpool names and limit must be valid. |
| Source | Maintained by product templates. Use bedit to implement a different value. |
| Applicability | BATCH |

Example: `com.splwg.grid.distThreadPool.threads.DEFAULT=5`

com.splwg.batch.scheduler.daemon - Scheduler Daemon

| | |
|---------------|---|
| Parameter | <code>com.splwg.batch.scheduler.daemon</code> |
| Description | Configures whether the threadpool should execute the online scheduler daemon. The online scheduler daemon checks that the Batch Submission transaction and initiates execution of <i>pending</i> batch jobs. |
| | <i>Note: This setting can be overridden using the -d option on the threadpoolworker[.sh] command line.</i> |
| | <i>Note: This setting should only be set to true in non-Production environment where online submission is enabled.</i> |
| | <i>Note: In any non-Production environment that uses online submission; one threadpool should have this setting enabled. Typically this is the DEFAULT threadpool but can be any other threadpool.</i> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Maintained by product templates. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>com.splwg.batch.scheduler.daemon=false</code> |

jmx.remote.x.access.file - JMX Access Control

| | |
|---------------|---|
| Parameter | <code>jmx.remote.x.access.file</code> |
| Description | Sets relative location of JMX access control file holding the valid user. Refer to JMX Security for more details. |
| Ordinality | Mandatory |
| Valid Values | Relative location of file from <code>\$\$SPLEBASE/%SPLEBASE%</code> . Defaults to <code>scripts/ouaf.jmx.access.file</code> |
| Source | Value inherited from templates. Use custom template to set this value to the desired location and file, if desired. |
| Applicability | WEB BAS |
| Example: | <code>jmx.remote.x.access.file=scripts/ouaf.jmx.access.file</code> |

jmx.remote.x.password.file - JMX Password Control

| | |
|-------------|---|
| Parameter | <code>jmx.remote.x.password.file</code> |
| Description | Sets relative location of JMX password file holding the valid passwords for <code>jmx.remote.x.access.file</code> . Refer to JMX Security for more details. |

| | |
|---------------|---|
| 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 template to set this value to the desired location and file, if desired. |
| Applicability | WEB BAS |
| Example: | <code>jmx.remote.x.password.file=scripts/ouaf.jmx.password.file</code> |

oracle.ouaf.jmx.remote.maxPortsToTry - Ports to Attempt to connect to

| | |
|---------------|--|
| Parameter | oracle.ouaf.jmx.remote.maxPortsToTry |
| Description | Number of ports to try starting from either oracle.ouaf.jmx.remote.startPort or spl.runtime.management.rmi.port . If oracle.ouaf.jmx.remote.startPort is not specified, the property spl.runtime.management.rmi.port is used that as the starting port. If neither property is defined or the maximum number of ports to try has been reached, an exception will be thrown that will prevent the system from starting. |
| Ordinality | Optional |
| Valid Values | Number. Default: 200 |
| Source | Set by BATCH_RMI_PORT . Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>oracle.ouaf.jmx.remote.maxPortsToTry=100</code> |

oracle.ouaf.jmx.remote.startPort - Override Batch JMX Port

| | |
|---------------|---|
| Parameter | oracle.ouaf.jmx.remote.startPort |
| Description | Alternative Port number allocated for Batch JMX facility . This value is typically the same as spl.runtime.management.rmi.port . But may be set to the Coherence JMX port if Coherence based JMX is used on a different port. |
| Ordinality | Optional |
| Valid Values | Valid JMX Port |
| Source | Set by BATCH_RMI_PORT . Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>oracle.ouaf.jmx.remote.startPort=6510</code> |

ouaf.batch.onlineLogDir - Online Line Batch Log Directory

| | |
|---------------|--|
| Parameter | ouaf.batch.onlineLogDir |
| Description | Sets the location of output from online batch submission and batch scheduler integration. If this property is set, batch logs will be accessible via the Batch Run Tree (application UI). Otherwise, the batch logs download link in the application will be disabled. |
| Ordinality | Optional |
| Valid Values | Valid location. Defaults to: \$SPLOUTPUT (%SPLOUTPUT% on Windows) |
| Source | Value inherited from BATCH_ONLINE_LOG_DIR from ENVIRON.INI . |
| Applicability | BATCH |
| Example: | <code>ouaf.batch.onlineLogDir=/tmp</code> |

ouaf.batch.onlineLogDir.retain – Retain Online Line Batch Log Files

| | |
|---------------|---|
| Parameter | ouaf.batch.onlineLogDir.retain |
| Description | Sets the total number of log files that are to be stored for each batch code and thread number. The log files for the particular batch code and thread number will be deleted except for the latest number of logs specified. |
| Ordinality | Optional |
| Valid Values | Number. |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>ouaf.batch.onlineLogDir.retain=3</code> The above example will keep the last three logs and delete the rest |

spl.runtime.batch.disableDeferredKeyInserts - Deferred Key Inserts

| | |
|--------------|--|
| Parameter | spl.runtime.batch.disableDeferredKeyInserts |
| Description | Whether the deferring of key insert in batches should be disabled or not. Enables (false) or disables (true) deferred Key Inserts. The default is to use deferred inserts. However if an aggressive key insert is desired in batches, a value of true should be used. <hr/> <i>Note: This setting is used by particular base batch jobs. Refer to the online documentation for individual batch jobs for more information.</i> <hr/> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Generated by template. Use bedit to implement a different value. |

| | |
|---------------|--|
| Applicability | BATCH |
| Example: | <code>spl.runtime.batch.disableDeferredKeyInserts=false</code> |

spl.runtime.batch.L2CacheMode - Threadpoolworker L2 Cache control

| | |
|---------------|--|
| Parameter | <code>spl.runtime.batch.L2CacheMode</code> |
| Description | <p>Defines the 2nd-level cache mode for threads in threadpool. Valid values are READ_ONLY (default), READ_WRITE or OFF.</p> <ul style="list-style-type: none"> • READ_ONLY is for performance as it loads the configuration entities into the 2nd-level (JVM-level) cache and accesses them from there only. In READ_ONLY mode, the cached entities may not be updated. • READ_WRITE also loads them into the L2 cache, but they can be updated. The drawback is that an update of a cached entity hits both the database and cache as it gets updated in both places. • In OFF mode the 2nd-level cache is completely disabled. Some jobs, for example LDAP import (F1-LDAP), need to update some of these entities (e.g. User), so it requires the L2 cache to be either OFF or READ_WRITE. The choice between OFF and READ_WRITE depends on requirement; the effect is the same. <hr/> <p><i>Note: This setting can be overridden using the -T2 option on the threadpoolworker[.sh] command line.</i></p> |
| Ordinality | Mandatory |
| Valid Values | [READ_ONLY READ_WRITE OFF] |
| Source | Maintained by product templates for threadpoolworker. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>spl.runtime.batch.L2CacheMode=READ_ONLY</code> |

spl.runtime.management.connector.url.default - Batch JMX URL

| | |
|---------------|--|
| Parameter | <code>spl.runtime.management.connector.url.default</code> |
| Description | <p>JMX URL for Batch JMX interface.</p> <hr/> <p><i>Note: This URL is generated and should not be altered.</i></p> |
| Ordinality | Optional |
| Valid Values | Valid JMX URL |
| Source | Generated by template |
| Applicability | BATCH |
| Example: | <code>spl.runtime.management.connector.url.default=service:jmx:rmi:///jndi/rmi://{host}:{port}</code> |

[/oracle/ouaf/batchConnector](#)

spl.runtime.management.rmi.port - Batch JMX Port

| | |
|---------------|---|
| Parameter | spl.runtime.management.rmi.port |
| Description | Port number allocated for Batch JMX facility . <i>Note: This setting can be overridden using the -i option on the threadpoolworker[.sh] command line.</i> |
| Ordinality | Optional |
| Valid Values | Valid JMX Port |
| Source | Set by BATCH_RMI_PORT . Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>spl.runtime.management.rmi.port=1001</code> |

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: <ul style="list-style-type: none"> • Caching is disabled at all layers of the architecture. This can be reenabled manually. • Screen Preloading is disabled. This can be reenabled manually. • Compression is disabled at all layers of the architecture. This can be reenabled manually. • The product is set to expanded mode to allow developers access to individual JEE objects rather than using WAR/EAR files. <p>It is highly recommended that this setting should be set to false in Production environments.</p> <p>This setting is only set to true where the Oracle Utilities SDK is used directly.</p> |
| Ordinality | Mandatory |
| Valid Values | [true false] |
| Source | Derived from WEB_ISDEVELOPMENT parameter from ENVIRON.INI . |
| Applicability | WEB BAS IWS |
| Example: | <code>spl.runtime.options.isDevelopmentMode=false</code> |

tangosol.coherence.distributed.localstorage - DistributedCache Storage

| | |
|---------------|---|
| Parameter | tangosol.coherence.distributed.localstorage |
| Description | Specifies whether this member of the DistributedCache service enables local storage. <hr/> <i>Note: This setting is used by the Oracle Coherence DistributedCache Service.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | [true false] |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.localport.adjust=true</code> |

tangosol.coherence.distributed.threads - Internal Cache Threads

| | |
|---------------|--|
| Parameter | tangosol.coherence.distributed.threads |
| Description | Specifies the number of daemon threads used by the distributed cache service. <hr/> <i>Note: This setting is used by the Oracle Coherence DistributedCache Service.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | Valid thread numbers. Default: 0 |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.distributed.threads=4</code> |

tangosol.coherence.invocation.threads - Internal Invocation Threads

| | |
|---------------|---|
| Parameter | tangosol.coherence.invocation.threads |
| Description | Specifies the number of daemon threads to be used by the invocation service. If zero, all relevant tasks are performed on the service thread. <hr/> <i>Note: This setting is used by the Oracle Coherence InvocationService.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | Valid thread numbers. Default: 0 |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.invocation.threads=4</code> |

tangosol.coherence.localhost - Unicast Host Address

| | |
|---------------|---|
| Parameter | tangosol.coherence.localhost |
| Description | Host name for local host for Unicast address. <hr/> <i>Note: This setting applies to unicast or well known address implementations.</i> <hr/> <i>Note: This setting is used by the Oracle Coherence unicast-listener.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | Valid host name defined in hosts file and/or DNS. |
| Source | Generated by template (wka only). Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.localhost=myserver.example.com</code> |

tangosol.coherence.localport - Unicast Port

| | |
|---------------|---|
| Parameter | tangosol.coherence.localport |
| Description | Port number for local host for Unicast address. <hr/> <i>Note: This setting applies to unicast or well known address implementations.</i> <hr/> <i>Note: This setting is used by the Oracle Coherence unicast-listener.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | Valid port number. |
| Source | Generated by template (wka only). Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.localport=6570</code> |

tangosol.coherence.localport.adjust - Unicast Port Adjustment

| | |
|---------------|--|
| Parameter | tangosol.coherence.localport.adjust |
| Description | Specifies whether the unicast port is automatically incremented if the specified port cannot be bound to because it is in use. <hr/> <i>Note: This setting applies to unicast or well known address implementations.</i> <hr/> <i>Note: This setting is used by the Oracle Coherence unicast-listener.</i> <hr/> |
| Ordinality | Optional |
| Valid Values | [true false] |
| Source | Generated by template (wka only). Use bedit to implement a different value. |
| Applicability | BATCH |

Example: `tangosol.coherence.localport.adjust=true`

tangosol.coherence.role - Threadpoolworker Role

| | |
|---------------|--|
| Parameter | tangosol.coherence.role |
| Description | This contains the name of the member role. This name allows an application to organize members into specialized roles, such as cache servers and cache clients. The name is also useful for displaying management information (for example, JMX) and interpreting log entries. |
| | <i>Note: This setting is used by the Oracle Coherence member-identity.</i> |
| Ordinality | Optional |
| Valid Values | Role Name |
| Source | Generated by template. Use bedit to implement a different value. |
| Applicability | BATCH |
| Example: | <code>tangosol.coherence.role=OUAF_Base_TPW</code> |

coherence-cache.config.xml

The `coherence-cache.config.xml` file is generated using the `coherence-cache.config.xml.template` located in the `templates` directory.

The [format and structure](#) are setup for the implementation of Oracle Coherence for the batch infrastructure within the product.

Note: This file is fixed in format and should not be altered unless instructed by Oracle Support.

tangosol-coherence-override.xml

The `tangosol-coherence-override.xml` file is generated using the templates located in the `templates` directory.

There are four templates for this file:

| Template | Contents |
|--|--|
| <code>tangosol-coherence-override.mc.be.template</code> | Multicast Overrides (for bedit) |
| <code>tangosol-coherence-override.ss.be.template</code> | Single Server Overrides (for bedit) |
| <code>tangosol-coherence-override.wka.be.template</code> | WKA Overrides (for bedit) |
| <code>tangosol-coherence-override.xml.template</code> | General Overrides |

The [format and structure](#) are setup for the implementation of Oracle Coherence for the batch infrastructure within the product.

Note: This file is fixed in format and should not be altered unless instructed by Oracle Support.