

Oracle® Insurance Policy Administration

Oracle Insurance Integration Accelerator for Oracle Insurance Policy Administration (OIPA) and Oracle Insurance Documaker

Installation Instructions

Version 9.4.1.0

Documentation Part Number: E23637_01

October 2011

Copyright © 2009, 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Where an Oracle offering includes third party content or software, we may be required to include related notices. For information on third party notices and the software and related documentation in connection with which they need to be included, please contact the attorney from the Development and Strategic Initiatives Legal Group that supports the development team for the Oracle offering. Contact information can be found on the Attorney Contact Chart.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Table of Contents

OVERVIEW	5
CUSTOMER SUPPORT	5
1. INSTALLATION	5
Installing the Extension	5
Locating Configuration Files	6
<i>OSC Properties</i>	6
<i>Service Registry</i>	6
Configuring the Template Store	6
<i>File-based Storage</i>	6
<i>Database-based Storage</i>	6
Basic File Layout	7
2. CONFIGURATION	7
SERVICE DEFINITION	7
<i>General Service Configuration</i>	8
<i>SOAP Service</i>	8
<i>File Service</i>	9
<i>JMS Service</i>	9
MESSAGE TEMPLATE	10
EXTENSION INVOCATION	10
<i>Transaction Math Extension (TYPE="PROCESS")</i>	10
3. SUPPORTED PLATFORMS	12
Application Servers	12
Database Storage	12
JMS Providers	12
4. CACHING	13
Configuring Coherence	13
Configuring the LRU Cache	13
Disabling the Cache	14
5. EXAMPLES	15
Push Notification Scenario	15
Service Definition	15
Template Definition	15
Extension Invocation	15
Alternative Delivery - File	16
Alternative Delivery - JMS	16

APPENDIX.....	17
A1. OSC PROPERTY FILE VALUES	17
A2. SERVICE REGISTRY TAGS.....	18

OVERVIEW

The Oracle Insurance Integration Accelerator leverages the Oracle Insurance Policy Administration's Extensibility Framework to provide a generic, reusable approach to outbound data delivery.

The extension provides mechanisms to specify delivery targets and message templates. The latter allows users to map between the data at a given extension point and the downstream interface. Further, it avoids the need to develop individual extensions for each downstream interface, while maintaining application performance and throughput.

CUSTOMER SUPPORT

If you have any questions about the installation or use of our products, please visit the My Oracle Support website: <https://support.oracle.com>, or call (800) 223-1711. When speaking with the support team the product association is with the Oracle Policy Administration system.

PRE-REQUISITES

Before you can begin to use the Integration Accelerator both the Oracle Insurance Policy Administration system and Documaker product must be installed.

1. INSTALLATION OF INTEGRATION ACCELERATOR

There are several steps to installing and configuring the Oracle Insurance Integration Accelerator and its resources.

Installing the Extension

It is shipped as a collection of jars: the primary osc-[version].jar and its dependencies. These will need to be added to the OIPA deployable.

Releases are available through eDelivery, in the OIPA Media Pack. Releases obtained through other means may not be supported.

Locating Configuration Files

There are two core files that need to be placed locally on the machines hosting the OIPA application. Values and settings for these files will be referenced throughout the documentation. Complete references for both can be found as appendices to this document. Refer to the below sections for more details on **OSC Properties** and **Service Registry**.

OSC Properties

The osc.properties file contains settings relevant to how the Integration Accelerator will execute. This file is generally configured at the beginning of a project and will rarely be updated.

The file needs to be located on the JVM classpath. This is typically achieved by adding it to the OIPA properties directory or creating an Integration Accelerator specific directory and adding it to the classpath. Once this file is configured, it can generally be deployed to all environments without modification.

Service Registry

The service-registry.xml contains a list of service definitions that are used. This file is updated as new services are added. The location of this file is defined in the OSC Properties. Set services.file equal to the path to this file.

As this file contains environment specific data, migration between environments should be manually reviewed. Typically there will be different endpoint instances serving Dev, QA, Production, etc. and this file abstracts those concerns from the rest of the configuration. Refer to the **How to Guide for OIPA to Documaker** for more details.

Configuring the Template Store

The Integration Accelerator can retrieve message templates from either the local file system or a database. The local file system is generally a simpler and faster option. The database may be a better solution for multi-cluster machines or where access to local storage is restricted.

File-based Storage

To enable file-based storage, a directory will need to be created to hold the templates on the local file system. This directory will then be specified in the osc.properties.

1. Create a directory to hold the templates
2. Add template.source=file to the osc.properties
3. Add freemarker.templateDir equal to the directory from step 1 to the osc.properties

Database-based Storage

To enable database-based storage, a table needs to be created in a database and a jdbc resource to reach that database needs to be created. These resources are then specified in the osc.properties file.

OSC Database Schema

```
CREATE TABLE OSCTemplate(
  TemplateID varchar2(40) NOT NULL,
  TemplateData CLOB NOT NULL,
  UpdatedGMT timestamp NOT NULL,
  primary key (TemplateID)
```

)

1. Create the table using the above schema.
2. Create a JNDI bound JDBC resource named OSCDS pointing to the database.
3. Add template.source=db to the osc.properties file.
4. Optionally, configure the template.minReadInterval to restrict how often the plugin will check for a new version of the template in the database. Default is 60000, i.e. 10 minutes.

Basic File Layout

Using a file based template store, most users will find the following to be the most convenient deployment structure.

OSC File Layout

/opt/Oracle/oipa (a.k.a. OIPA_Home)

```
| -osc
|   |-service-registry.xml
|   |-properties
|   | -osc.properties
|   |-templates
```

The properties directory is added to the JVM classpath entry. Templates are stored in the templates directory which is specified in the osc.properties file.

2. CONFIGURATION

Configuring a call using the Integration Accelerator involves three configuration steps:

1. A service endpoint needs to be configured in the service registry. The endpoint identifies the technical details of where the message will be delivered and the template that should be used to create the message. This definition also contains the service Id that will be used elsewhere to identify the service. Refer to Service definition section below for more details.
2. A message template will need to be created. The template is responsible for formatting the data appropriately for the type of service. The Integration Accelerator implicitly requires the message be text, as that is what is produced by the template engine. Certain service types may have additional requirements (namely, a web service endpoint only accepts valid XML). Refer to Message Template section of this document for more details.
3. OIPA will need to be configured to call the Integration Accelerator extension and pass in data. Refer to Extension Invocation section of this document for more details.

SERVICE DEFINITION

There are several supported service types: SOAP services for invoking SOAP endpoints over HTTP, file services for writing text to the filesystem, and jms services for delivering text to JMS endpoints. Each type of service requires different information to perform its function, but all services are defined in a common registry.

General Service Configuration

The root tag of the service-registry.xml file is <Services>. There are no XML namespaces used.

Each service is defined using a <Service> tag in the service-registry.xml file. There are several elements common to every service definition.

Node	Required	Description
/Services/Service/@id	Yes	Used to identify service by the configuration. Must be unique.
/Services/Service/@type	Yes	Defines the service type. Valid values are soap, file, and jms.
/Services/Service/TemplateName	Yes	Defines the template key for this service.

SOAP Service

A SOAP service allows the delivery of XML messages to a SOAP endpoint over HTTP. The associated template is responsible for defining the contents of SOAP Body, while the service manages construction of the SOAP message itself in accordance with a provided WSDL.

The service definition is required to provide a WSDL and identify the service and port to be invoked. There are several additional options that allow for overriding default parameters.

Node	Required	Description
/Services/Service/WSDLLocation	Yes	Specifies the URL location of the WSDL for the service.
/Services/Service/ServiceName	Yes	Specifies the name of the service (defined in the WSDL) to be called. Must be a valid QName.
/Services/Service/ServicePort	Yes	Specifies the port of the service (defined in the WSDL) to be called. Must be a valid QName.
/Services/Service/ServiceLocation	No	Overrides the service location in the WSDL with the value specified. Must be a valid URL.
/Services/Service/SOAPAction	No	Overrides the SOAP Action specified in the WSDL. <i>See note below before including this tag.</i>
/Services/Service/IgnoreResponse	No	Ignores response from service and returns success immediately. Valid options are true and false. Default is false.
/Services/Service/SecurityType	No	Can be used to override WS-Security or WS-Policy information.

QName Format

The QName format is the text equivalent of Java QName class. It specifies the Qualified Name of an element in a document. QName is printed as {Namespace URI}Local Part. Also note, that in the case of a WSDL, the Namespace URI is, in fact, the target namespace of the document, not the WSDL's namespace.

WSDL Unavailable?

If a WSDL is not published for a service (or not available to the OIPA system online) it can be provided locally. Simply specify the WSDL URL location using a file:// prefix instead of http://.

SOAP Action Issues

The presence of the SOAP Action header in a SOAP message has meaning. Some services will expect to receive no header, some will expect a blank header, and some may expect a value.

For an *absent* SOAP Action header, neither the WSDL nor the service definition can include *any* reference to it. For a *blank* SOAP Action header, provide the <SOAPAction> tag with no value in the service definition. In general, you can ignore this tag and omit it from the service-registry.xml

Waiting for a Response

If the <IgnoreResponse> tag is provided and set to true, then the Integration Accelerator will not wait for a return value from the service. This means several things from an execution perspective.

- The plugin will still generate errors if there are issues opening a connection with the service or if a transmission error occurs while sending the message.
- The result of successful transmission will always be <Success/>.

The plugin will not generate errors if the remote system was unable to understand or process the message. There can be no assurance that the message was successfully received with a true value.

File Service

A file service allows for the delivery of any textual data to the server's file system. In general, this should be leveraged with a remote file sharing protocol (NFS, Samba, etc.) or a scheduled FTP task to collate files between servers.

The service definition is required to provide a directory where files should be written. There are additional tags to control the naming of files and replacement handling.

Node	Required	Description
/Services/Service/Directory	Yes	The directory where files will be written.
/Services/Service/FileNameType	No	The type of naming scheme to use. Valid options are Random, Reference, and Simple. The default value is Random.
/Services/Service/FileName	No	For Random this element has no effect. For Reference, it is the name of a variable in the context that holds the desired file name. For Simple, it is the name of the file.
/Services/Service/ReplaceExisting	No	If true then a file of the same name as the requested output name will be overwritten. If this is false an existing file of the same name will cause an error. Default is false.

Reports

Using a simple file name and "ReplaceExisting" makes it easier to export batch reports to the file system as part of Plan or Company level processing. Don't forget to ensure output names are unique amongst reports.

JMS Service

A JMS service allows for the delivery of any textual data to JMS mapped destination (queue or topic).

A connection factory and destination are both required for delivery. Optional requirements include the QoS specifications for time to live and priority. These values should map to their relevant JMS values.

Node	Required	Description
/Services/Service/ConnectionFactory	Yes	The JNDI name of the JMS Connection Factory.
/Services/Service/Destination	Yes	The JNDI name of the JMS Destination.
/Services/Service/TimeToLive	No	The message life time in milliseconds. (Unlimited-0)
/Services/Service/Priority	No	The priority of the message. (Low-0, High-9, Default-4)

MESSAGE TEMPLATE

The Integration Accelerator operates using a template system for messages as opposed to XSLT. There are several key advantages to this approach. It avoids the overhead of building an intermediate XML format from the data and it allows the template to interact directly with the data structures from OIPA.

Message templates are used to define the outgoing data payload. Payload formats differ depending on the type of delivery used. SOAP delivery requires a valid XML document. File and JMS delivery support any text format.

The message template language currently used in the Integration Accelerator is FreeMarker. The FreeMarker [Template Author Guide](#) and [Syntax Reference](#) cover all of the functionality available.

The extension is responsible for exposing data to FreeMarker so it can be used in template processing, however, data is derived from the OIPA extension point in use. For the Math Transaction extension, all elements passed as parameters are exposed to the template by their NAME attribute. More information about passing data to the Integration Accelerator can be found in Extension Invocation Section below.

EXTENSION INVOCATION

Navigation

Once a service is defined and the message template is constructed, the extension needs to be configured in OIPA. There are several configurable extension points in OIPA.

Transaction Math Extension (TYPE="PROCESS")

The math extension point relies on creating a new MathVariable of type PROCESS in a transaction rule in OIPA and then setting the following values to initiate the extension call.

Node	Required	Description
MathVariable/@VARIABLENAME	Yes	The request variable name
MathVariable/@TYPE	Yes	Must be "PROCESS"
MathVariable/@NAMESPACE	Yes	Must be "com.oi.osc"
MathVariable/@OBJECT	Yes	Must be "MathPlugin"
MathVariable/@DATATYPE	Yes	Must be "OBJECT"

Once the call is configured, MathVariables from the current execution can be passed into the plugin. Note that the parameter name specified, maps to the name used in the message template and the Parameter element's text, maps to the MathVariable name in the transaction.

Node	Required	Description
MathVariable/Parameters/Parameter/@NAME	Yes	Defines the variable reference used in the template.
MathVariable/Parameters/Parameter/@TYPE	Yes	Valid values are INPUT and OUTPUT.
MathVariable/Parameters/Parameter	Yes	Defines the name of the MathVariable to be passed to the plugin.

OIPA Variable Datatypes

The message template language provides full support for interacting with complex objects. As such, it is highly recommended that arrays and maps be passed directly to the template and manipulated there.

The Integration Accelerator requires three specific parameters be defined in addition to any required for data mapping.

Parameter Name	Parameter Type	Parameter Value
service.id	INPUT	The name of service to be invoked. Corresponds to the id attribute of the <Service> definition.
returnValue	OUTPUT	The name of a predefined TEXT variable to hold the return value from the service.
errorValue	OUTPUT	The name of a predefined TEXT variable to hold any possible error value that may be returned from the service.

Output Values

If any error arises during processing, the text of the error will be placed in the errorValue. The configuration should check to ensure this variable is blank before continuing processing.

3. SUPPORTED PLATFORMS

Application Servers

The Integration Accelerator relies heavily on tools and libraries provided by the Java environment. In general, any JEE 5+ application server with at least a 1.6 JVM should provide adequate support. The following is the list of tested/known

Application Server	Version	Works?	Notes
Oracle WebLogic	11g R1+	Yes	Supported with all patch levels.
Redhat JBoss	4.x	No	Not Supported
Redhat JBoss	5.x, 6.x	Maybe	Should work, but untested.
IBM WebSphere	6.1	Maybe	Should work, but untested.
IBM WebSphere	7.0	Maybe	Pending OIPA issue resolution.

Database Storage

With this version introduced the ability to store templates in a database. Our usage is very light and anything supported by JPA will work. For practical reasons, only databases certified with OIPA will be supported by Integration Accelerator.

JMS Providers

In theory any JMS provider should work correctly. Below is the list of tested configurations. If there is a client-based desire for any additional configurations, same can most likely be accommodated.

JMS Provider	Version	Works?
WebSphere MQ	7.0	Yes
WebLogic Built-in	11g R1+	Yes
Oracle Service Bus	11g+	In process

4. CACHING

The Integration Accelerator leverages two kinds of caching during execution: a local cache and a distributed cache. These both operate through the same interface, but semantically work differently. The local cache supports non-serializable objects whereas the distributed cache does not.

By default, the Integration Accelerator uses a local LRU cache and Coherence as a distributed cache. If Coherence support is not desired, the LRU cache can be substituted for the distributed cache as well.

Configuring Coherence

The Coherence configuration for Integration Accelerator needs to be merged with the configuration for OIPA. This can be done by adding the following to the coherence-cache-config.xml file.

Coherence Configuration for Integration Accelerator

```
<!-- Map Templates to Distributed Near Scheme -->
<cache-scheme-mapping>
  <cache-mapping>
    <cache-name>OSC_Templates</cache-name>
    <scheme-name>OSCScheme</scheme-name>
  </cache-mapping>
</cache-scheme-mapping>

<!-- OSC Distributed In-memory Cache -->
<cache-schemes>
  <local-scheme>
    <scheme-name>SampleMemoryScheme</scheme-name>
  </local-scheme>
  <distributed-scheme>
    <scheme-name>OSCScheme</scheme-name>
    <backing-map-scheme>
      <local-scheme>
        <scheme-ref>SampleMemoryScheme</scheme-ref>
      </local-scheme>
    </backing-map-scheme>
  </distributed-scheme>
</cache-schemes>
```

This creates the necessary relevant configuration. The configuration can be further adjusted to meet deployment specific needs.

Configuring the LRU Cache

The LRU cache is a simple built-in solution for caching non-serializable objects. It is configured entirely through the osc.properties file.

Property Name	Value Type	Default	Notes
cache.lru.<cache_name>.initSize	Integer	100	For a given LRU sub-cache, specify the initial size
cache.lru.<cache_name>.maxSize	Integer	1000	For a given LRU sub-cache, specify the maximum size.

Where the <cache_name> is the internal sub-cache Id (currently only OSCJMS and OSC_Templates).

Disabling the Cache

Either (or both) of the caches can be disabled by overriding the default cache implementation. A special no-op cache has been provided, which stops the cache from using any data.

To change, set `cache.distributed.type` or `cache.local.type` to `com.oj.osc.cache.NoCache`.

5. EXAMPLES

Push Notification Scenario

Company ABC wants to push information to an external service whenever a new policy is submitted. The service expects an XML file like:

Sample XML

```
<PolicyNotify>
  <PolNum>Policy Number</PolNum>
  <Amount>Policy Face Amount</Amount>
</PolicyNotify>
```

Service Definition

First, define the service in the service-registry.xml.

Service Registry Definition

```
<Service id="policyNotify" type="soap">
  <TemplateName>PolicyNotify.ftl</TemplateName>
  <WSDLLocation>http://www.abc.com/services/PolicyNotify?wsdl</WSDLLocation>
  <ServiceName>{http://www.abc.com/xml/PolicyNotify/}PolicyNotifyService</ServiceName>
  <ServicePort>{http://www.abc.com/xml/PolicyNotify/}PolicyNotifyServicePort</ServicePort>
</Service>
```

This defines a service with the policyNotify id as a SOAP service. The WSDL specified must be accessible to the application server at runtime. The service name and port are taken from the WSDL and identify the component to be invoked.

Template Definition

Second, define the message template.

Message Template - PolicyNotify.ftl

```
<PolicyNotify>
  <PolNum>${PolicyNumber}</PolNum>
  <Amount>${FaceAmount}</Amount>
</PolicyNotify>
```

This defines the message template. The template name specified in the <TemplateName> tag of the service description must match the name of the file.

Since this is a template, the file should strongly resemble the downstream format. Variables are introduced where content from OIPA is needed.

Extension Invocation

Third, call the extension from the OIPA config.

OIPA Transaction Configuration

```
<MathVariable VARIABLENAME="ServiceID" TYPE="VALUE"
DATATYPE="TEXT">policyNotify</MathVariable>
<MathVariable VARIABLENAME="ReturnValue" TYPE="VALUE" DATATYPE="TEXT"></MathVariable>
<MathVariable VARIABLENAME="ErrorValue" TYPE="VALUE" DATATYPE="TEXT"></MathVariable>

<MathVariable VARIABLENAME="PolicyNotify" TYPE="PROCESS" NAMESPACE="com.oi.osc"
OBJECT="MathPlugin" DATATYPE="OBJECT">
  <!-- Required Parameters for every call -->
  <Parameter NAME="service.id" TYPE="INPUT">ServiceID</Parameter>
  <Parameter NAME="returnValue" TYPE="OUTPUT">ReturnValue</Parameter>
  <Parameter NAME="errorValue" TYPE="OUTPUT">ErrorValue</Parameter>
  <!-- Data Parameters -->
  <Parameter NAME="PolicyNumber" TYPE="INPUT">PolicyNumber</Parameter>
  <Parameter NAME="FaceAmount" TYPE="INPUT">FaceAmount</Parameter>
</MathVariable>
```

This sample defines three new text variables to hold values for the call. One for the service id and two for the return values. It assumes that PolicyNumber and FaceAmount are already defined prior to this definition.

Alternative Delivery - File

If Company ABC wants to leverage file delivery instead of SOAP, the only configuration that needs to change is the service definition.

Policy Notify - File

```
<Service id="policyNotify" type="file">
  <TemplateName>PolicyNotify.ftl</TemplateName>
  <Directory>/opt/interfaces/policyNotify</Directory>
  <FileNameType>Reference</FileNameType>
  <FileName>PolicyNumber</FileName>
</Service>
```

This specifies the directory that should be used for output as well as some optional settings. The filename type specifies that the filename is a reference to a provided value, in this case PolicyNumber. Since the directory is specific to the interface this should be alright. ReplaceExisting could also be added to allow for overwriting the file should the activity be reprocessed.

Alternative Delivery - JMS

If Company ABC wants to leverage JMS delivery instead of SOAP, the only configuration that needs to change is the service definition.

Policy Notify - JMS

```
<Service id="policyNotify" type="jms">
  <TemplateName>PolicyNotify.ftl</TemplateName>
  <ConnectionFactory>jms/ConnectionFactory</ConnectionFactory>
  <Destination>jms/Destination</Destination>
</Service>
```

JMS delivery only requires a ConnectionFactory and Destination reference. These are both JMS artifacts that are configured in the application server and bound to a jvm or cluster. If the destination is shared between interfaces consider setting the Priority flag to rank messages if important.

APPENDIX

A1. OSC PROPERTY FILE VALUES

The following table is a complete list of all the valid property keys for the osc.properties file. Full descriptions of these properties and their meaning can be found elsewhere in this document. In most cases, default values should not need to be changed unless they are required.

Property Name	Value Type	Required	Default	Notes
cache.distributed.type	Class implementing Cache	No	com.oi.osc.cache.CoherenceCache	
cache.local.type	Class implementing Cache	No	com.oi.osc.cache.LRUCache	
cache.lru.<cache_name>.initSize	Integer	No	100	For a given LRU sub-cache, specify the initial size.
cache.lru.<cache_name>.maxSize	Integer	No	1000	For a given LRU sub-cache, specify the maximum size.
factory.constructor	Class implementing IConstructor	No	com.oi.osc.constructor.FreeMarkerConstructor	
factory.security	Class implementing ISecurityProvider	No	com.oi.osc.security.WalletSecurityProvider	
freemarker.templateDir	String	Maybe	n/a	Required if loading templates from filesystem. Must specify full path to template directory.
services.file	String	Yes	n/a	Must specify full path to service registry.
template.minReadInterval	Integer	No	60000	Value in milliseconds.
template.source	{ file, db }	Yes	n/a	User must specify how templates will be loaded.
wallet.path	String	Maybe	n/a	If using the WalletSecurityProvider, this must specify a path to

Property Name	Value Type	Required	Default	Notes
				the Oracle Wallet file.
wallet.key	String	Maybe	n/a	If using the WalletSecurityProvider, this must specify the wallet password.

A2. SERVICE REGISTRY TAGS

Navigation

List of all the possible service registry tags. Full descriptions of these tags and their meaning can be found elsewhere in this document.

All Node listings are relative to <Services>/<Service>.

The column Relevance endeavors to indicate when the tag is relevant to a service definition. The column Required then indicates when whether the tag is required in that particular case.

Node	Value Type	Relevance	Required	Notes
@id	String	All	Yes	Must be unique to the service registry.
@type	{ soap, file, jms }	All	Yes	
TemplateName	String	All	Yes	Must match a valid template file name or database key.
WSDLLocation	URL	soap	Yes	URL of the SOAP service's WSDL.
ServiceName	QName	soap	Yes	QName of the service to invoke.
ServicePort	QName	soap	Yes	QName of the service port to invoke.
ServiceLocation	URL	soap	No	Overrides value in WSDL.
SOAPAction	String	soap	No	Do not include tag unless specifically required.
IgnoreResponse	Boolean	soap	No	
Username	String	soap	No	
PasswordKey	String	soap	No	
SecurityType	{ HTTP, XWSS }	soap	No	
XWSSConfig	String	soap: xwss	Yes	Path to XWSS policy file to apply to service.
Directory	String	file	Yes	Absolute path to directory for files.
FileNameType	{ Random, Reference, Simple }	file	No	Defaults to Random.
FileName	String	file: reference, simple	No	Specifies a reference to a variable with the file name or the file name itself depending on FileNameType.

Node	Value Type	Relevance	Required	Notes
ReplaceExisting	Boolean	file: reference, simple	No	Specifies whether to overwrite an existing file. Default: false.
ConnectionFactory	String	jms	Yes	JNDI name of JMS Connection Factory
Destination	String	jms	Yes	JNDI name of JMS Destination
TimeToLive	Integer	jms	No	Time to live, in seconds. 0 = unlimited. Default = 0
Priority	Integer [0-9]	jms	No	Message priority, low to high. Default = 4