ORACLE
INSURANCE

Oracle Health Insurance Back Office

Service Consumer Installation & Configuration Manual

Version 1.4

Part number: F27899-01

March 18, 2020

CHANGE HISTORY

Release	Version	Changes
10.19.1.3.0	1.0	• Creation
10.19.1.4.0	1.1	• Small textual adjustments
10.19.2.0.0	1.2	• No changes, republished with a new part number
10.20.1.0.0	1.3	• No changes, republished.
10.20.2.0.0	1.4	• Added more information about calls that exceed time-outs • Added calling consumers in custom code

RELATED DOCUMENTS

A reference in the text (**doc[x]**) is a reference to another document about a subject that is related to this document.
Below is a list of related documents:

Doc[1]	OHI Back Office - Installation & Configuration Manual (CTA13508)
Doc[2]	OHI Back Office - Custom Development for Oracle Health Insurance Back Office (CTA13678)

Contents

1 Introduction

The OHI Back Office web service consumers are designed to call third-party web services or web services that have an interface that has been defined by OHI. These web services are present in the world outside OHI Back Office and are called from the PL/SQL code within the OHI Back Office database, with several intermediary steps.

Each web service consumer has a corresponding PL/SQL wrapper package providing a PL/SQL interface to the public methods of the web service consumer.

The PL/SQL wrapper functions are called from within the OHI Back Office database and make use of two database queues with JMS payload. OHI provides an optional Oracle Service Bus (OSB) project definition, to implement the communication with the external service.

1.1 Licenses

No additional OHI Back Office license option is required to use the web service consumers. The web services (incoming service calls) and web service consumers (outgoing service calls) together constitute the 'Service Layer'. The web services part of the 'Service Layer' DO require an additional OHI license.

For further information, please consult your OHI sales representative.

For the underlying technology, Oracle Fusion Middleware and/or the Oracle Service Bus, you have to acquire separate technology licenses, these are not included within OHI Back Office.

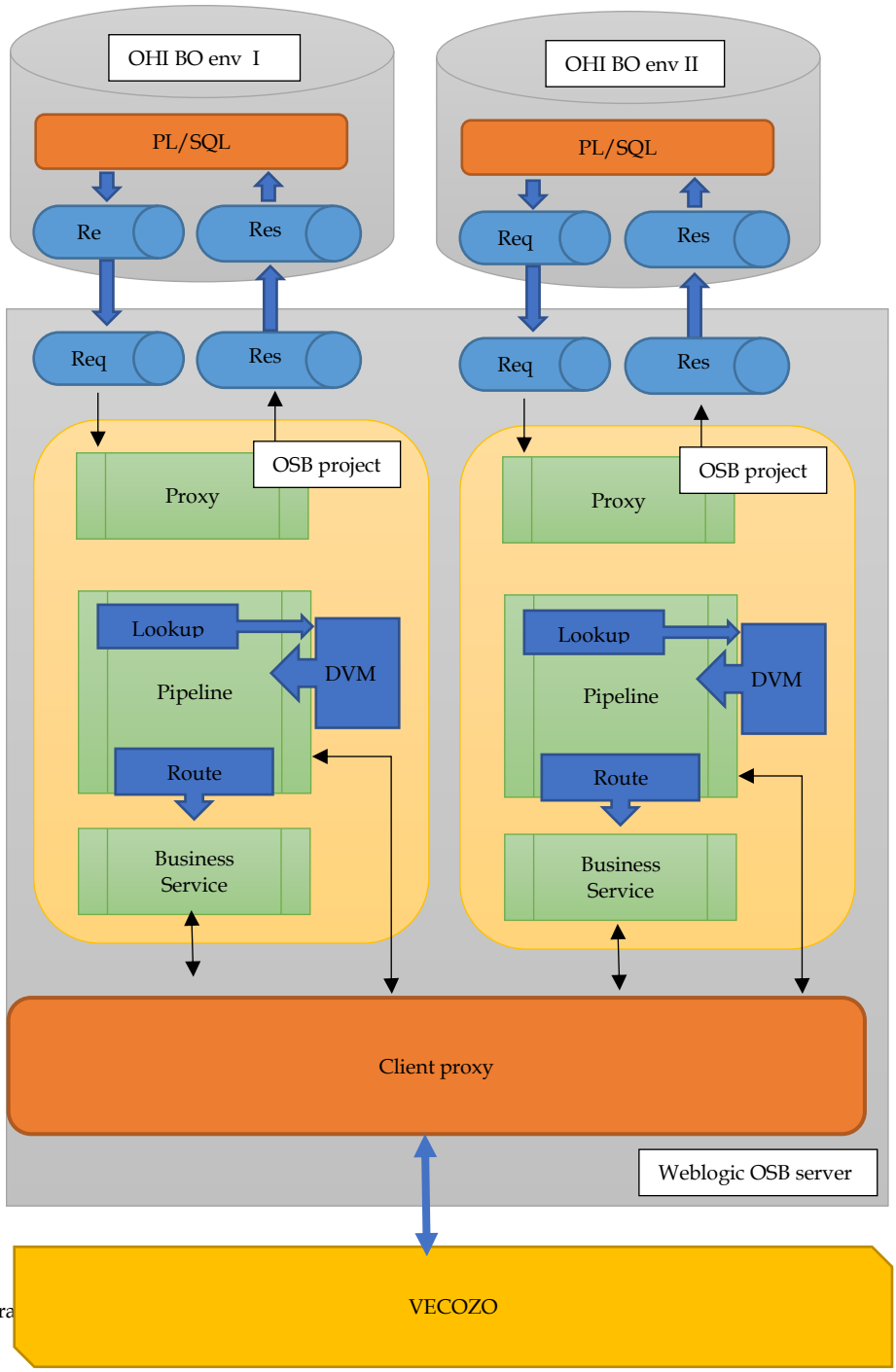
2 Architectural overview

The service consumer functionality as described afterwards offers synchronous calls to external web services. This means OHI calls to the web service are ended when the call returns a response or a time-out is reached.

This chapter gives a high level architectural overview of the web service consumers, in short 'SVC'. The abbreviation SVC is used in many OHI module names to make clear the files belong to the **S**er**v**ice **C**onsumer implementation, which makes use of 2 database queues with a JMS payload . These queues can be exposed by implementing an OSB project definition delivered by OHI. Customers are at liberty to expose the database queues and process the messages in their own custom way.

2.1 Design

The diagram below shows how two separate OHI Back Office environments use one Oracle Service Bus (OSB) environment with two OHI BO specific OSB project deployments to call web services offered by Vecozo (as an example) through a customer-specific proxy layer called by the OSB projects.



The solution is based on the following architecture:

- The OHI Back Office application puts a message of a specific type on the request queue from within a PL/SQL routine in the database.
- The calling PL/SQL routine behaves 'synchronously': after the message has been put on the request queue, the calling process waits for a response message on the response queue before it proceeds.
- Internally the PL/SQL routine stores a (SOAP) request message on an Advanced Queuing (AQ) database queue with a JMS payload.
- In the Oracle Service Bus, running within WebLogic Server, the AQ queue in the database is accessible by the WebLogic Server as a JMS queue, because it has been defined within WebLogic as a Foreign Queue. Enqueued requests are immediately dequeued by an OSB Adapter process which passes on the request to the designated web service or proxy service. The response is received and placed back on a JMS response queue, which is a Foreign Queue definition mapped to the AQ response queue in the database.
- The calling PL/SQL routine has been waiting for the response message to appear on the response queue since it has enqueued the request message. Technically, it has started a dequeue for a message with the corresponding correlation id. When the response is received the result is handed over as parameter to the caller. At that moment the PL/SQL call that implements the service call has finished.

2.2 Main components

This paragraph describes the components as shown in the diagram. These components are needed for the service consumer implementation to work.

2.2.1 Queues

The queues in the database are standard database Advanced Queues (AQ) with a JMS (Java Message Service) based payload of type AQ$_JMS_TEXT_MESSAGE. They are created by the OHI installation software, no additional action is required.

The AQ$JMS_TEXT_MESSAGE is a structure that conforms to the JMS standard with plain text as message body (payload). This text can be formatted in any structure (XML, json, fixed length etc.)

The retention time on both queuesis configured as one day. This means that the payload of the message will remain on the queue for a day after the message has been processed.

The customer can increase the retention time, providing the contents of the queue does not grow to much. The number of messages in the queue does affect the performance.

The message on the request queue gets a default expiration time of 25 seconds, consisting of the time-out for the call plus 10 seconds . The message will be put on the error queue if the message is not processed (dequeued) within this time. This

expiration time can be overruled for each service consumer by a Back Office parameter setting.

The same two queues, with database names ALG_SVC_REQUEST_QUEUE and ALG_SVC_RESPONSE_QUEUE, are used for all service consumer implementations.

The type of web service being called is identified by the JMS type property of the request message. This property is used in the Oracle Service Bus project to determine which endpoint to call.

2.2.2 Weblogic

In Weblogic the request and response queue are exposed in a so called 'foreign server' with its own connection factory in a JMS module, supported by a connection pool with a data source that connects to the OHI BO database. In this way the database queues can be used by the Oracle Service Bus as standard JMS queue. The actual implementation as database based queue is not visible for the processes accessing the JMS queus.

2.2.3 Oracle Service Bus project

As part of the service consumer implementation, OHI delivers an Oracle Service Bus project definition. It is not mandatory to use this. Customers are free to implement analternative custom solution as long as it handles the request and response queue messages in a similar way. See the Custom Development Guide **Doc[2]** for details on how to develop your owm message handling.

An Oracle Service Bus project usually consists of at least

- a 'proxy service', which is the receiver of a request call
- a 'pipeline' definition implementing the actual logic
- a 'business service' where the endpoint is configured.

The OSB project uses a JMS Adapter as 'proxy service' to process messages enqueued on the request queue. The 'pipeline' determines which 'business service' to call to process the request message. The business service processes the service call and puts the response message on a separate response queue. A correlation id, which is part of the JMS payload, is used to 're-route' the response back on the response queue to the caller The calling process starts a dequeue with this correlation id directly after having enqueued the request.

Customers need to run multiple OHI BO environments with potentially different OHI BO releases. A different OHI BO release may have a different version of the OSB Project, e.g. when a change is needed in the payload of a consumer service.

The OSB Project comes with tooling to support the parallel deployment of multiple OSB Projects in a single OSB instance. Names of OSB objects can include the OHI environment name, to make them recognizable and unique.

The OSB project must also support the addition of new service consumer with a minimum of code changes, and be configurable at deployment time.

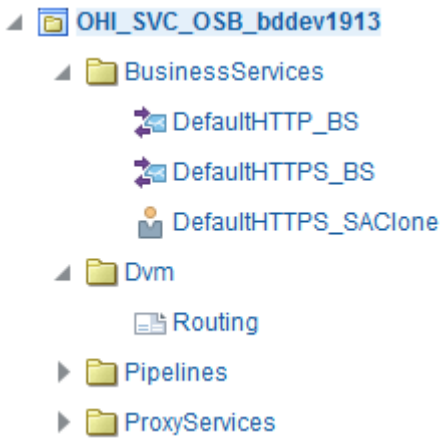
To support this, a 'dynamic routing configuration' has been implemented. A Domain Value Mapping (DVM) file is used. The DVM is a component which is new since release 12 of the OSB. It supports a 'domain value' table/list, in fact a simple table structure.

The pipeline determines the correct endpoint based on the JMS type in the request message and the corresponding entry in the DVM In this way, a different end point is chosen for each service consumer. The DVM file can be filled with the correct end points for the specific OHI environment as part of the deployment.

Below a few records are shown in the DVM, after deployment. These are records of the OHI environment specific OSB project file for the environment 'bddev1913'. The project name is 'OHI_SVC_OSB_bddev1913' and is visible in the naming of the RouteService column value. The JMSType column indicates the different service consumers.

JMSType	RouteType	RouteService	endPointURI
SVL1001C	BS	OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS	http://slc11ali.us.oracle.com:8088/SVL1001C
SVL1002C	BS	OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS	http://slc11ali.us.oracle.com:8088/SVL1002C
SVL1003C	BS	OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS	http://slc11ali.us.oracle.com:8088/SVL1003C
SVL1004C	BS	OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS	http://slc11ali.us.oracle.com:8088/SVL1004C

The BusinessService DefaultHTTP_BS is part of the same project as shown below in the opened project tree:



Based on the Route Type values in this DVM table a message can be routed in one of three possible ways. Note that the Routing Type can differ for each service consumer:

- When Route Type contains 'BS' a Business Service of the same project (by default) is specified that represents the provider system. In practice, this provider will not be the external party (e.g. VECOZO), but a service on an internal application server (e.g. in a DMZ) that forwards the call to the external party, e.g. VECOZO. The Business Service endpoint (that points to the provider) is also set according to the DVM record, using the 4th column.There is a default HTTP business service as well as an HTTPS business service. The protocol of the endPointURI should match that of the Business Service (e.g. be http or https too).
  There is also an option to route a request to a different business service as part of a different OSB project implementation, for example, if there are some specific configuration requirements, such as transformation of the message payload, additional security, etc.

- The Routetype column may also contain PP to specify a Pipeline to implement custom logic, if required. This requires a PipeLine definition to be added to the OSB project which makes adoption of newer versions of the OHI provided OSB project harder. The endPointURI is not relevant in this situation.

- Another option is to specify PS to redirect the request to a different Proxy Service that may be implemented by the customer, This Proxy Service might be used to handle message translation and interfacing with provider system(s). In this situation the endPointURI is not relevant either.

The 'business service' definition contains calls to external web services as well as service accounts containing the client certificates or credentials to implement connectivity to client proxy services. The OSB currently includes one http and one https enabled business service, as described in the first bullet.

3 Prerequisites

The functionality of the service consumers may not be required in all OHI BO environments. It is needed in the Production environment and in some test environments. In environments where the service consumers are needed, this functionality must be enabled by fulfilling the prerequisites described below.

The deployment itself is described in the next chapter.

3.1 Database account for providing queue access

The optional database account OHI_JMS_QUEUE_USER needs to be present. This account is also needed for accessing other JMS queues provided by OHI Back Office, so it may already be present.

How to configure this account is described in the generic OHI Back Office Installation, Configuration and DBA Manual (as mentioned in Doc[1]).

When the account is present it is possible to directly access the database queues from outside the database as JMS payload queues. In Java the JMS API is available by means of the oracle.jms Java package. For more information please use the database documentation and consult the Advanced Queuing User's Guide and/or the Advanced Queuing Java API Reference.

3.2 WebLogic Server environment for providing JMS queue access

If you decide not to use the Oracle Service Bus project definition provided by OHI, but use a custom solution to dequeue messages in the request queue, to implement the web service call and to return the response message on the response queue, you may still want to access the database queues as standard JMS queues.

One way of doing is, using WebLogic, is to set up a datasource, connection pool, JMS Server and JMS Module to create Foreign Queue definitions for both database queues. These Foreign Queues will then act as standard JMS queues.

You may choose to use a separate WebLogic domain for these Foreign Queues, or re-use a domain that is already used for the OHI web service components SVL, HSL or PSL

For smaller OHI BO test environments with a limited use it may be convenient to combine these Foreign Queues with other service deployments in the same domain. It may well be that you configure Foreign Queues for multiple environments in the same domain.

A separate, clustered domain may be more appropriate fora highly available production situation.

As this all is standard WebLogic Server functionality available in all recent releases, there is no strict requirement for the WebLogic Server release to use to implement the Foreign Queues. A supported and certified version for the database release of OHI Back Office will suffice. Make sure you applyrecent security and corrective patches on WebLogic.

3.3 Oracle Service Bus environment

If you decide to use the Oracle Service Bus project definition that is delivered as part of the OHI release you need to have a separate WebLogic Server installation for running the Oracle Service Bus (OSB).

In this way you can apply patches separately on the OSB technology stack and for the WebLogic Server environment that is used for the OHI Back Office web services and Forms components. OHI will perform a separate certification for the OSB technology stack, so the OSB and Forms/Web Services WebLogic Server versions will not have to be upgraded at the same moment.

Consult the Oracle Service Bus documentation for installing the WebLogic Server software and the Oracle Service Bus.

When the OSB is running you need to create Foreign Queues and accompanying components for each OHI BO environment that needs to be serviced by the OSB installation. Typically an OSB project and a set of Foreign Queues is created per OHI BO environment.

OHI advises a separate highly available OSB instance for a production situation.

4 Deployment Instructions

This chapter explains

- how to create the Foreign Queue definitions to access the two database queues
- how to deploy the OSB project delivered as part of OHI BO releases
- how to tailor the OSB project to your situation
- how to support multiple OHI BO environments with multiple instances of the OSB project The versions of the OSB projects can be different.

Each OHI BO environment needs the following basic WebLogic objects, before the OSB project can be imported. If you decide not to use the OSB project but do want to use the Foreign JMS Queues to expose the database queues, these objects are needed too, with the exception of the Work Manager.

- A data source that can access the relevant request and response queue for the service consumer implementation messages.
- WebLogic foreign JMS queues for each of these queues.
- A work manager. The OSB project will link the work manager to the queues.

The OHI BO release contains template scripts and template property files to automate the setup of the Foreign Queues and a working OSB environment. The instructions in this chapter are based on these templates.

As these are templates you are free to change them as long as you do realize that a newer OHI releases may deliver a changed version of a template and you may have to merge changes into your version. The templates only provide limited functionality for simple deployment situations and are provided as a starting point. They are not intended as a configuration tool for handling more complicated configuration requirements. If you run into situations that are not supported adapt the templates to the situation or use a different method to deploy the components.

The templates are delivered in directory $OZG_BASE/Back-Office when OHI BO is patched so you will find them on the application server for OHI Back Office. Your OSB installation may well reside on other application servers. The scripts need to be run on the OSB server, so you need to consider where you will store and maintain them

NOTE: in this chapter, we will use “bddev1913” as the name of the OHI Environment

4.1 Secure storing of credentials

If you do not want to specify unencrypted usernames and passwords in scripts and do not want toprovide them over and overwhen running the scripts interactively, consider the option tostore the Weblogic credentials for the administrative account (typically ‘weblogic’) of the OSB domain in a set of secure files.

The commands below save the credentials for a Weblogic user in two files (a credential file and an accompanying key file).

The credential file can be decrypted by WLST using the key file during the execution of the WLST scripts.

To create the files execute the following steps:

- Set the envirionment variables to the WebLogic domain for OSB (typically $DOMAIN_HOME/setDomainEnv.sh)

- Run WLST:
  `$MW_HOME/oracle_common/common/bin/wlst.sh`

- Connect to the admin server:
  connect('weblogic', '<pw>', 't3://<host>:<port>')

- Create the files with `storeUserConfig`:
  ```
  storeUserConfig(userConfigFile='myconfigfile.secure',
  userKeyFile='mykeyfile.secure')
  ```

  The files are created in the current directory unless you include a directory in the values.

- Exit:
  ```
  disconnect()
  exit()
  ```

In a property file, e.g. file `ohi_svc_osb_import.properties` that is discussed later in this chapter, provide both files (assuming the file location is in `/ohi/security`):

```
admin.configfile=/ohi/security/myconfigfile.secure
admin.keyfile=/ohi/security/mykeyfile.secure
```

4.2 Foreign Queue configuration

This paragraph describes how to define Foreign Queues to access the two database queues. For most non production environments the scripted configuration will facilitate the setup and provide an efficient deployment.

For a highly available production environment a manual setup may be preferred. We expect you to have the requisite knowledge to make this decision and to set this up. Feel free to tailor the Foreign Queue definitions to your needs to support a potential high load, fail-over, etc..

The instructions in this paragraph assume you are deploying to the OSB domain. If you are not using the OSB project, the Foreign JMS Queues may be deployed on another WebLogic Domain. The instructions are valid in that situation too. You may not require the Work Manager in that case.

Open a terminal window on the application server where you want to deploy the Foreign JMS Queues.

NOTE: It is possible to connect to the Admin Server of a WebLogic Domain remotely. That does work for the deployment of the Foreign JMS Queue, but not for the deployment of OSB Project later on In the remainder of this document, we will assume you run the deployment actions on the application server where the Admin Server of the OSB domain runs .

Please follow the configuration instructions below:

- Set the envirionment variables to the WebLogic domain for OSB (typically $DOMAIN_HOME/setDomainEnv.sh)

- Determine a working directory on the OSB server. directoryConsidercreating this directory in a central locationdirectory outside the $OZG_BASE if you plan to adapt hem or use them for multiple OHI BO environments. Be aware that the files in $OZG_BASE/conf/Back-Office will be overwritten by OHIPATCH.

- Copy the following files from $OZG_BASE/conf/Back-Office to the working directoryThe first file is the script that will be actually called while the other three create the datasource, the foreign queues and the work manager, as implied by their name.
  `ohi_svc_wls.sh`
  `ohi_svc_wls_datasource.py`
  `ohi_svc_wls_queues.py`
  `ohi_svc_wls_workmanager.py`

- Copy `ohi_svc_wls.properties.template` to file `ohi_svc_wls.properties` (or include the name of the OHI environment e.g. `ohi_svc_wls.bddev1913.properties`) in that same working directory.

- Adapt the file `ohi_svc_wls.properties` to your specific circumstances.

- A short explanation for each property is given:

    - admin.url:
      Insert the administrative URL of the admin server (prefix it with t3 or t3s depending on whether you use the normal or SSL port).

    - admin.configfile and admin.keyfle:
      Insert the config- and keyfile name and location to connect securely automatically or leave these blank and provide username and password at the prompt at runtime.

    - env.name:
      Specify the name of the OHI BO environment (typically used also for the $OZG_BASE directory). Beware of the case sensitive nature of the value.

    - env.domain:
      Specify the WebLogic OSB domain of the OSB server (or the WebLogic domain if not deploying to OSB).

    - env.target.type:
      Specify the target type forthe WLS data source and foreign queues. Choices are *Server* for a single Managed Server or *Cluster* for a cluster of Managed Servers.

    - env.target.name:
      Specify the name of the Server or Cluster.

    - datasource.user:
      This is the first property of a set of properties that are used to create the datasource definition in WebLogic to access the queues in the

OHI BO database of the OHI environment. The value MUSTbe OHI_JMS_QUEUE_USER.

- datasource.pwd_osvar:
  This property specifies the name of an OS environment variable that may contain the value for the password of the datasource user account. This is one way to specify the password. See the description of file `ohi_svc_wls.sh` described below.

- datasource.ohi_table_owner:
  This specifies the table owner account of the OHI BO tables (formost OHI customers OZG_OWNER). datasource.dbhost:
  The server name where the OHI BO database runs. This will be used to compose the classical JDBC connect string for the datasource, consisting of server name, port name and service name.. If a different format is required the provided scripts could be adapted .

- datasource.dbport:
  For the host specified in the previous property please provide the port of the Oracle Net listener (most often 1521).

- datasource.dbservice:
  specify the service name toconnect to the database through the listener on the specified database server.

NOTE: You have 3 options to supply the password for ohi_jms_queue_user. This is the user for the data source.

1. **Supply the password interactively, while running `ohi_svc_wls.sh`**
2. **Register the name of an environment variable <XXX> in datasource.pwd_osvar in file `ohi_svc_wls.properties` and set <XXX> with this value before starting `ohi_svc_wls.sh`. Unset <XXX> after the script has been executed**
3. **Store the password in file `ohi_svc_wls.sh` before running the script and remove it after the script has been executed.**

- Optionally adapt the file `ohi_svc_wls.sh` and set the password for the JMS queue database user account (ohi_jms_queue_user). When not set (no value specified) it will be asked interactively when running the script.

- Run the file `ohi_svc_wls.sh` while specifying the properties file. This will create the different components in WebLogic to create the Foreign Queue definitions:
  `./ohi_svc_wls.sh ohi_svc_wls.properties`

- Use the Weblogic Admin Console of the OSB domain at http://<server>:<port>/console (Not the OSB Console at http://<server>:<port>/servicebus) to check that the script has created the objects correctly:

Path	Name
Services -> Data Sources	DS_OHI_SVC_bddev1913
Services -> Messaging -> JMS Servers	OHI_SVC_bddev1913_Server
Services -> Messaging -> JMS Modules	OHI_SVC_bddev1913_Module

Services -> Messaging -> JMS Modules -> OHI_SVC_bddev1913_Module	OHI_SVC_bddev1913_ForeignServer
Services -> Messaging -> JMS Modules -> OHI_SVC_bddev1913_Module -> OHI_SVC_bddev1913_ForeignServer -> Destinations	OHI_SVC_bddev1913_REQUEST_QUEUE
Services -> Messaging -> JMS Modules -> OHI_SVC_bddev1913_Module -> OHI_SVC_bddev1913_ForeignServer -> Destinations	OHI_SVC_bddev1913_RESPONSE_QUEUE
Environment -> Work Managers	OHI_SVC_bddev1913_MT_2
Environment -> Work Managers	OHI_SVC_bddev1913_WM

- Restart the Managed Server(s) that form the Target.

4.3 OSB project deployment

The next step is to deploy the OHI OSB project.

OHI delivers a simple generic OSB project that needs to be customized and configured for each OHI environment.

The following steps are needed to create and deploy an environment specific project:

- Copy the following files from the Back-Office templates directory ($OZG_BASE/conf/Back-Office) to the working directory you created earlier.
    - `ohi_svc_osb_configure.properties.template;` copy this to `ohi_svc_osb_configure.properties` or to `ohi_svc_osb_configure.bddev1913.properties`
    - `ohi_svc_osb_import.properties.template;` copy this to `ohi_svc_osb_import.properties (t`his file is not OHI environment specific, but will differ for each OSB domain)
    - `ohi_svc_osb_configure.sh`
    - `ohi_svc_osb_import.sh`
    - `ohi_svc_osb_import.py`
- Copy the file `OHI_SVC_OSB.jar` from `$OZG_BASE/java` directory to the same working directory.
- Adapt the file `ohi_svc_osb_configure. bddev1913.properties` to your situation.
    - environmentCode: typically you enter the name you use to identify the OHI BO environment, like you did before. We use `bddev1913` here.

- jms* properties:
  Use the same replacements as in the previous paragraph “Foreign Queue configuration”. The results should match the properties of the objects in the WebLogic Admin Console.
- Next there is a set of three properties for each service consumer that is defined in OHI BO:
  - …ROUTE_TYPE:
    The type of the OSB service to route toThis can be BS for a business service, PS for a proxy service and PP for a pipeline. See the paragraph “Oracle Service Bus project” above for an explanation.
  - …ROUTE_SERVICE :
    Provide the default business service value for the property.When using unencrypted http this is: OHI_SVC_OSB/BusinessServices/DefaultHTTP_BS. When using encrypted https this is: OHI_SVC_OSB/BusinessServices/DefaultHTTPS_BS
  - …URI:
    Specify the endpoint where the business service can be reached. The protocol (hhp or https) must match the ROUTE_SERVICE.

- Adapt the file `ohi_svc_osb_import.properties:`
  - admin.url : Specify the same value as in `ohi_svc_wls.properties.`

  admin.configfile and admin.keyfle:
  Insert the config- and keyfile name and location to connect securely automatically or leave these blank and provide username and password at the prompt at runtime. Use the same values as in `ohi_svc_wls.properties.`See paragraph “Secure storing of credentials” for details.Run the configuration script that modifies the standard OSB project definition file to an environment specific project definition:
  ```
  ./ohi_svc_osb_configure.sh
  ohi_svc_osb_configure.bddev1913.properties
  OHI_SVC_OSB.jar
  ```

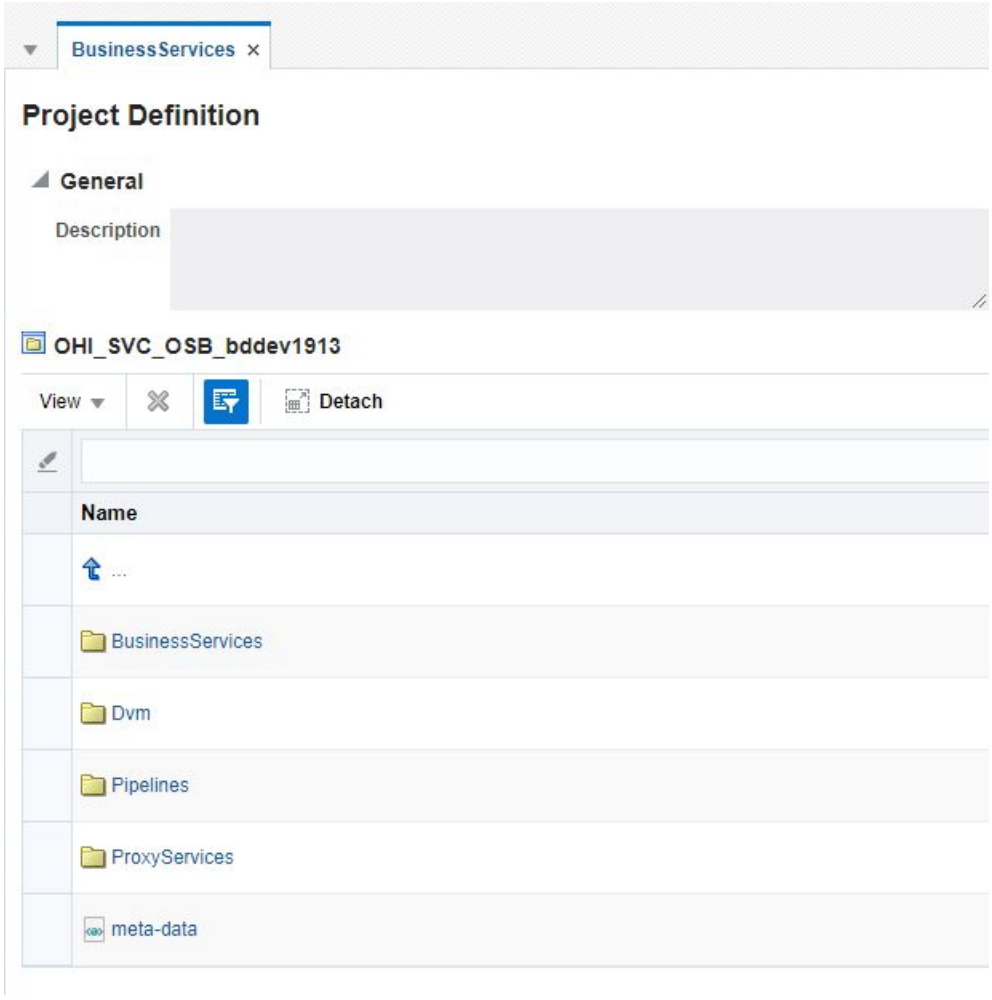
- As a result you should have a new project .jar file in a subdirectory ‘output’ of the working directory with the name `OHI_SVC_OSB-bddev1913.jar` .

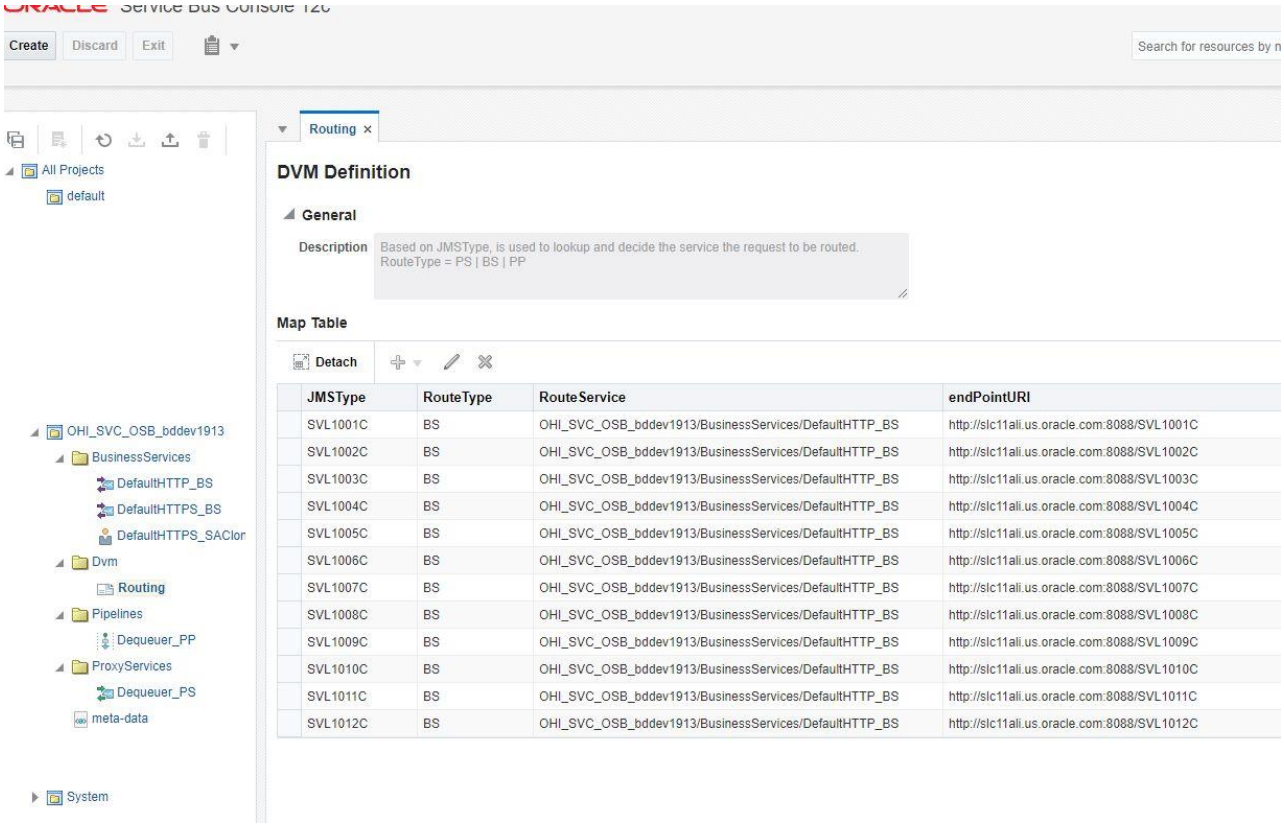
- Copy or move the new . jar file one level up to your working directory.

- Make sure you set your environment to the WLS OSB settings.

- Run the OSB import script to import the environment specific project definition (for environment bddev1913 in the example below):
  ```
  ./ohi_svc_osb_import.sh ohi_svc_osb_import.properties
  OHI_SVC_OSB-bddev1913.jar
  ```

- Use the OSB Console at http://<server>:<port>/servicebus) to check that the script has created the objects correctly:


BusinessServices
Project Definition
General
Description
OHI_SVC_OSB_bddev1913
View
Detach
Name
...
BusinessServices
Dvm
Pipelines
ProxyServices
meta-data


Create
Discard
Exit
Search for resources by n
All Projects
default
OHI_SVC_OSB_bddev1913
BusinessServices
DefaultHTTP_BS
DefaultHTTPS_BS
DefaultHTTPS_SAClor
Dvm
Routing
Pipelines
Dequeuer_PP
ProxyServices
Dequeuer_PS
meta-data
System
Routing
DVM Definition
General
Description
Based on JMSType, is used to lookup and decide the service the request to be routed.
RouteType = PS | BS | PP
Map Table
Detach
JMSType
RouteType
RouteService
endPointURI
SVL1001C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1001C
SVL1002C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1002C
SVL1003C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1003C
SVL1004C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1004C
SVL1005C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1005C
SVL1006C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1006C
SVL1007C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1007C
SVL1008C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1008C
SVL1009C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1009C
SVL1010C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1010C
SVL1011C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1011C
SVL1012C BS OHI_SVC_OSB_bddev1913/BusinessServices/DefaultHTTP_BS http://slc11ali.us.oracle.com:8088/SVL1012C

5 Mapping web service consumers

To actually call the web services in the outside world, you need to supply an intermediate mapping. This will map the OHI definition of the web service contract to the similar, but slightly different, web service contract of the external web service.

The OHI specific definition typically differs in namespace and may miss some functionality that is not implemented or may have a slightly different structure to provide the functionality.

When comparing the OHI specific contracts and the external versions the mapping should be quite trivial and easy.

5.1 Mapping/transformation

To do the runtime transformation for the web service consumers, an application server or middleware tier is a requirement. This will map the OHI specific WSDL to an outside world WSDL. That middleware tier will implement the actualcall, including the required authentication, to the service in the outside world.

The mapping or transformation can easily be implemented through service bus functionality.

Please load the .wsdl files for this purpose in your proxy or service bus environment. These .wsdl files can be obtained at the service provider organization, e.g. VECOZO.

The OHI specific WSDL files can be found in $OZG_BASE/xml on the application server instance of the environment. The files have names like SVLxxxxC.wsdl where xxxx can be any number between 0000 and 9999.

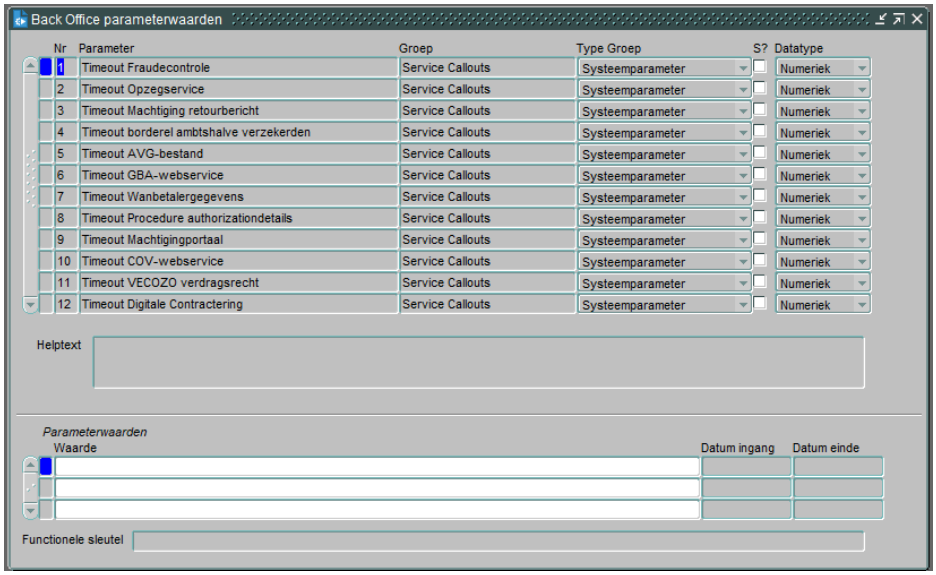
Currently the files SVL1001C.wsdl up to SVL1012C.wsdl do exist.

6 Back Office configuration

Part of the configuration is done in the OHI Back Office application.

6.1 Time-out setting and behaviour

A time-out value per service consumer can be set in the Back Office parameter values screen (SYS1145F) of the application instance. All parameters can be found in the group 'Service callouts', for each individual web service that can be consumed:



This specified time-out value determines the maximum time the OHI calling process waits for a response message to appear on the response queue, before returning an error message.

The default value which is used when not set through this parameter for a specific service is 15 seconds.

Note: The value set for these parameters is a maximum for the time-out. Setting a high value could cause a screen or batch session seem to "hang" waiting for the response during at the most the specified nr of seconds.

What is the effect of this setting?

- The expiration time of the request is based on this time-out. Some seconds after the time-out is passed the request message will be put on the error queue
- After the time-out is passed the calling procedure will stop and return an error. Responses placed after this exceeded time-out will not be processed. It is advised if the response is placed by custom code on the response queue to also provide an expiration time on this response message.

Note: The value of a Back Office parameter is cached when called for the first time. It is advised to stop and start the batch scheduler after changing the time-out for services handled by the business event framework like the procedure authorization details service to reinitialize this cached value.

6.2 Activation

To enable the OHI Back Office processes to use the queues instead of the deprecated database callouts used prior to release 10.19.1.3.0 the Back Office parameter values for the endpoints needs to be set to the fixed text <USE_QUEUE>.

Group VECOZO

- EVR URL request handler for service SVL1001C
- Premieachterstand URL request handler for service SVL1002C
- Opzegservice URL request handler for service SVL1002C
- Machtigingen URL request handler for service SVL1003C
- AVG URL request handler for service SVL1005C
- Ambtshalve URL request handler for service SVL1004C
- Wanbetaler URL request handler for service SVL1007C
- COV URL request handler for service SVL1010C
- EESSI URL request handler for service SVL1011C
- VDC URL request handler for service SVL1012C

Group GBA

- GBAEndPoint for service SVL1006C

Group SERVICE CONSUMER

- Procedure Auth URL request handler for services SVL1008C and SVL1009C

As it may be difficult to relate these English names to the Dutch prompts which are typically used in the user interface you can query the OHI Back Office database with the query below to find the relevant parameters per service:

```
select bop.naam
,      bop.prompt
,      bop.datatype
,      bop.ind_actief
,      boe.waarde_number
,      boe.creatie_moment
from   alg_back_office_parameters bop
join   alg_bo_parametergroepen bog
  on   ( bop.bog_id = bog.id )
left outer join alg_bo_parameterwaarden boe
  on   ( boe.bop_id = bop.id )
where bog.code = 'Service Callouts'
order  by bop.naam
```

7 Calling service consumers in custom code

The majority of the service consumers are being called by the OHI Back Office processes. However some of them can or should be called from within a custom code process.

This chapter contains a few directions when a service consumer is being called from custom code like a policy acceptance check. Some examples can be found in **Doc[2]**.

7.1 Interface

Every service consumer has a PL/SQL package interface with the following naming convention: SVC_SVLxxxxC_AQ_PCK where xxxx is a four digit number.

Per service operation a procedure is available with at least a pi_request parameter as input and a po_response parameter as output.

The soap fault returned by the operation can be retrieved by calling the associated get function. If an operation has a specific fault defined than use this function. In all other situations, when these specific functions are not available, use the get_svc_fault which will be available in these situations as standard function.

8 Monitoring and Troubleshooting

This chapter contains some troubleshooting paragraphs. These can be used if the service consumer calls do not work and result in errors that are not functional by nature.

8.1 Monitoring

Apart from using the tools provided by Fusion Middleware Control and Weblogic , monitoring specific OHI Service Consumer encountered error situations could be useful to identify issues in communication with downstream providers or other issues. When an error happens during the OHI OSB service execution, it will be logged in the default Managed Server log of the OSB server (severity level ERROR), providing only minimal error information for security reasons. For example:

```
Message    [null, null, null, ERROR] OHISVL: Error during
service execution: <OHISVLErrorInfo>

Supplemental Detail
  <faultstring>OHISVL-100: OSB error during backend
invocation</faultstring>
  <JMSCorrelationID>10000000023906</JMSCorrelationID>
  <JMSType>SVL1003C</JMSType>
  <operation>IndienenTerugkoppeling</operation>
</OHISVLErrorInfo>
```

The log message would contain the static string “OHISVL: Error during service execution” and extra information within details to identify the issue.

There are two types of error scenarios that willbe logged:

- A SOAP fault is returned by provider system
- An error happens within the OHI OSB project pipeline layer for the service consumer implementation

Therefore, the faultstring in the log message would be either the corresponding faultstring value from the provider system or the OHI OSB error code.

These are the current error codes from the OHI SVC OSB project:

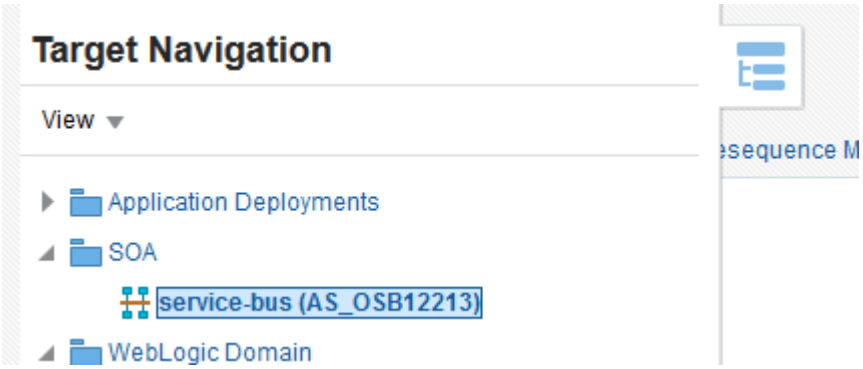
- OHISVL-100: OSB error during backend invocation
- OHISVL-101: Incomplete routing configuration for JMSType
- OHISVL-110: OSB technical error – Any runtime error that does not fall in the error cases above

8.2 Reporting

If more detailed information needs to be logged and traced, reporting can be enabled on the Dequeuer_PP pipeline service. Reporting should be off by default, for performance and privacy reasons. By using a CorrelationID value as a search key, for example, it is possible to find all the records during the processing of a particular message in the OSB layer. Report details will give more information on payload (the actual request message processed by the OSB and/or the actual response message that was received from the called web service), technical and transport settings, etc.

Message reports are accessible through the Enterprise Manager Fusion Middleware Control GUI (at http://<server:<port>/em)

Access to these reports can be obtained via the target navigation menu.
Select SOA and than service bus. The monitoring is available under the operations tab:



Operations

View ▾ Apply Revert

Name	Path	Type	State	Reports	Monitoring	Aggregation Interval
DefaultHTTP_BS	OHI_SVC_OSB_bddev1913/BusinessServices	Business Service	☑	...	☐	10 Mins
DefaultHTTPS_BS	OHI_SVC_OSB_bddev1913/BusinessServices	Business Service	☑	...	☐	10 Mins
Dequeuer_PP	OHI_SVC_OSB_bddev1913/Pipelines	Pipeline	...	☑	☐ (S)	10 Mins
Dequeuer_PS	OHI_SVC_OSB_bddev1913/ProxyServices	Proxy Service	☑	...	☐	10 Mins

If needed, Oracle Service Bus tracing functionality can be enabled in case the reporting feature does not provide enough information.

Note: After deployment verify that the reporting option is disabled globally or it is disabled for all the pipeline services of OHI OSB projects, to avoid generating unnecessary report information.

8.3 Querying Database Advanced Queues

Both request and response messages are temporarily stored in the database advanced queue tables ALG#SVC_REQUEST_TAB and ALG#SVC_RESPONSE_TAB. These tables also implement the exception/error queues.

These queue tables can be queried like any normal table, using the following query:

```
select q.corrid
,      q.state
,      q.expiration
,      q.enq_time
,      q.deq_time
,      q.user_data.header.type
,      q.user_data.text_vc
,      q.user_data.text_lob
,      q.q_name
```

```
from alg#svc_response_tab q
order by to_number(q.corrid) desc
```

- The corrid column contains the correlation id used to uniquely identify the specific message.
- The state defines the state the message is in. Typically you would see any of the values 0, 2 or 3.
    - 0; message is ready and available for dequeue
    - 2; message is processed and/or retained. Meaning it is dequeued but still present on the queue until the retention time has passed. That retention time is set on the queue.
    - 3; message is expired. The message is not dequeued within the expiration time set on the message.
- The expiration column holds the time in seconds before the message expires based on the enqueue time.
- Enq_time contains the timestamp when the message was put onto the queue.
- Deq_time contains the timestamp when the message was read from the queue by the consumer of the queue.
- The user_data column holds all the functional data of the message:
    - Header.type contains the JMS type of this specific message. It can be used to identify the type of message for this record (from which service consumer the message originates, like SVL1001C, etc.).
    - Either text_vc or text_lob holds the actual (SOAP) message. Based on the message size the database puts it in one of these two columns.
- The q_name column identifies to which queue the message belongs. It can be either the normal request or response queue or the exception queues (aq$_alg#svc_request_tab_e and aq$_alg#svc_response_tab_e) when the message is moved to the exception queue. The exception queue is also often referenced as the error queue.

Note: When monitoring the queues pay specific attention to:

- Messages with state 3. This may indicate that the time-out configured in the OHI Back Office parameters is too small
- Messages in the error queues; these could identify other issues.

8.4 Common errors

As discussed in the monitoring paragraph a small set of OHI OSB project errors can occur, apart from the errors that can occur from the provider side.

OHISVL-100 usually means that the endpoint which is set in the DVM is not accessible, available or not meant for the given message type.

OHISVL-101 means that one or more values in the DVM are incorrect or missing for the given service type.

When an ORA-3xxxx is given this usually means the response can not be validated against the stored xsd(s). There can be many reasons why the validation failed, from wrong namespaces to invalid values.

SVL-184 (SVL-00007) means a time-out has occurred. The response message was not available within the given time.

SVL-208 (SVL-00008) means some form of a SOAP fault is returned. The details returned within the SOAP fault may help in finding what actually went wrong.