Oracle Health Insurance Back Office

OHIJET Application Installation & Configuration Manual

Version 1.25

Part number: G24367-01

January 9, 2025

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.

ORACLE JET 9.2.5 License

```
# Oracle JET 9.2.5

You may not use the identified files except in compliance with the Universal Permissive License
(UPL), Version 1.0 (the “License.”)

You may obtain a copy of the License at https://opensource.org/licenses/UPL. A copy of the license
is also reproduced below.

Unless required by applicable law or agreed to in writing, software distributed under the License
is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.

See the License for the specific language governing permissions and limitations under the License.

Copyright I 2014, 2020 Oracle and/or its affiliates
The Universal Permissive License (UPL), Version 1.0


Subject to the condition set forth below, permission is hereby granted to any person obtaining a
```

copy of this software, associated documentation and/or data (collectively the “Software”), free of charge and under any and all copyright rights in the Software, and any and all patent rights owned or freely licensable by each licensor hereunder covering either (i) the unmodified Software as contributed to or provided by such licensor, or (ii) the Larger Works (as defined below), to deal in both

(10) the Software, and (b) any piece of software and/or hardware listed in the lrgrwrks.txt file if one is included with the Software (each a “Larger Work” to which the Software is contributed by such licensors), without restriction, including without limitation the rights to copy, create derivative works of, display, perform, and distribute the Software and make, use, sell, offer for sale, import, export, have made, and have sold the Software and the Larger Work(s), and to sublicense the foregoing rights on either these or other terms.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHANGE HISTORY

Release	Version	Changes
10.18.1.3.0	0.1	- Creation
10.18.1.3.0	0.2	- Revision 07-09-2018 - Add 'Back Office' to 'Prerequisites' - Revised 'Deploy HSL and PSL services'
10.18.1.3.0	0.3	- Revised PSL installation
10.18.1.3.0	0.4	- Revised architecture diagram - Minor textual changes - Revised note about hsl.tokenvalidation.rotor
10.18.1.3.0	1.0	- Reviewed. Slightly adjusted and updated to version 1.0.
10.18.2.0.0	1.1	- Republished with new part nr.
10.18.2.3.0	1.2	- Added description for PSL services deployment. - Added description of delivery of properties file templates.
10.19.1.0.0	1.3	- Added Oracle 6.1.0 JET License text - Renamed to OHIJET Application Installation & Configuration Manual - Revised and updated 'HSL and PSL services'
10.19.1.2.0	1.4	- Updated the list of required PSL services - Updated list of modules which require authorization
10.19.1.3.0	1.5	- Changed location of war files to $OZG_BASE/java - BAC0014J, BAC0016J and BAC0018J added in the list of services which might be deployed and in the list of modules which require authorization
10.19.1.4.0	1.6	- Introduced OHIJET.ear and adapted deployment instructions and references to individual web service deployments.
10.19.2.0.0	1.7	- No changes, republished with a new part number.
10.20.1.0.0	1.8	- Oracle JET version has changed.
10.20.3.0.0	1.9	- Oracle JET version has changed.
10.20.4.0.0	1.10	- Oracle JET version has changed.
10.20.6.0.0	1.11	- Oracle JET version has changed.
10.20.7.0.0	1.12	- Added an extra reference to **Doc[1]** for the deployment of HSL services (HSL_POL. HSL_REL, etc).
10.20.8.0.0	1.13	- Oracle JET version has changed.
10.21.1.0.0	1.14	- Oracle JET version has changed.. New part number. - Changed the Deployment validation paragraph.
10.21.2.0.0	1.15	- Browser certification info added in Deployment validation paragraph - Database Service name en Database Instance ID added to example in Deployment validation paragraph.
10.21.4.0.0	1.16	- Explicitly stated PSL_USER cannot be the same as HSL_USER
10.22.1.0.0	1.17	- Replaced token value and server names with placeholders. Republished with new part number.
10.22.8.0.0	1.18	- Added authentication with Oracle Access Manager.
10.23.1.0.0	1.19	- Info about certified browsers has moved to SecureSites. New part number.
10.23.6.0.0	1.20	- Psl.properties changed; obsolete properties removed.
10.23.6.0.0	1.21	- Added authentication with OpenID.
10.24.1.0.0	1.22	- No changes, republished with a new part number.
10.24.5.0.0	1.23	- Typing error corrected
10.24.8.0.0	1.24	- Services HSL_AUN and HSL_JUP have been moved to PSL_AUN and PSL_JUP - Updated references to the HSL installation manual because of the introduction of OAuth2 for the HSL services. - Restructured the documentation and removed duplicate information
10.25.1.0.0	1.25	- The PSL_AUZ webservice has been removed. References to the service in this document have been removed. - Published with a new part number.

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 HTTP Service Layer (HSL) Installation & Configuration Manual (docs.oracle.com)

Contents

1 Introduction

The OHI Back Office JET Application provides web modules for OHI Back Office users.

OHI Back Office web modules are designed from the ground up as productivity tools for power users.

Whereas most of the Forms modules help to navigate through the OHI Back Office data to support multiple potential different processes, the web modules are designed to support a specific process.

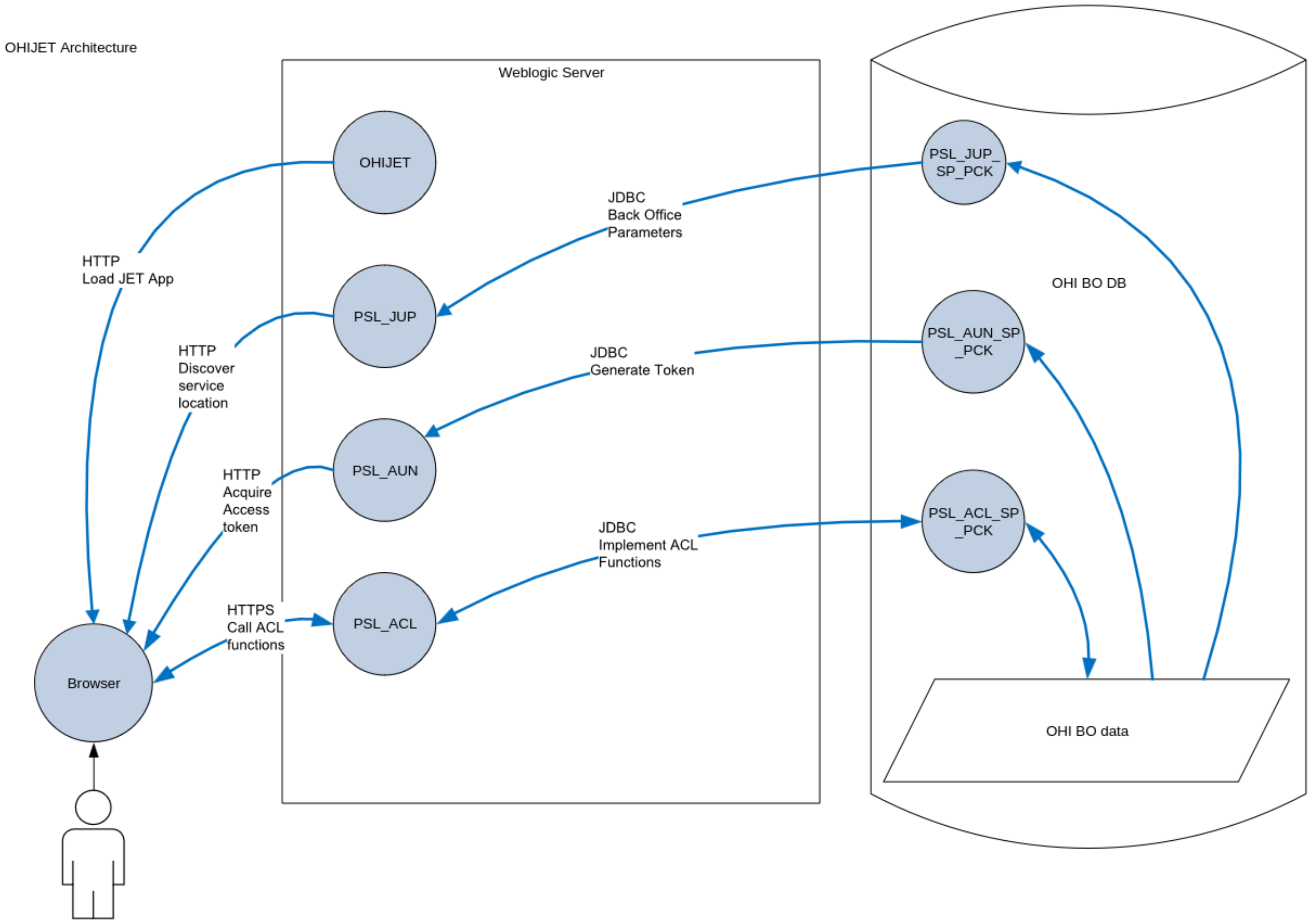
All web modules of the OHIJET application and their supporting components are packaged as a single EAR file: OHIJET.ear.

The OHIJET application is deployed to Oracle WebLogic Application Server (WLS).

This document describes how to install the OHIJET application and its related services.

2 Architectural overview

The diagram below shows the components of the OHI BO JET Application:



2.1 OHIJET Application

The OHIJET application was built using Oracle's JET toolkit for building modern web applications.

Among other things, Oracle JET contains UI components, data binding support to synchronize between the front-end application and a database back end and has mobile support.

You can find more information on

http://www.oracle.com/webfolder/technetwork/jet/index.html

The OHIJET application consists of multiple 'components':

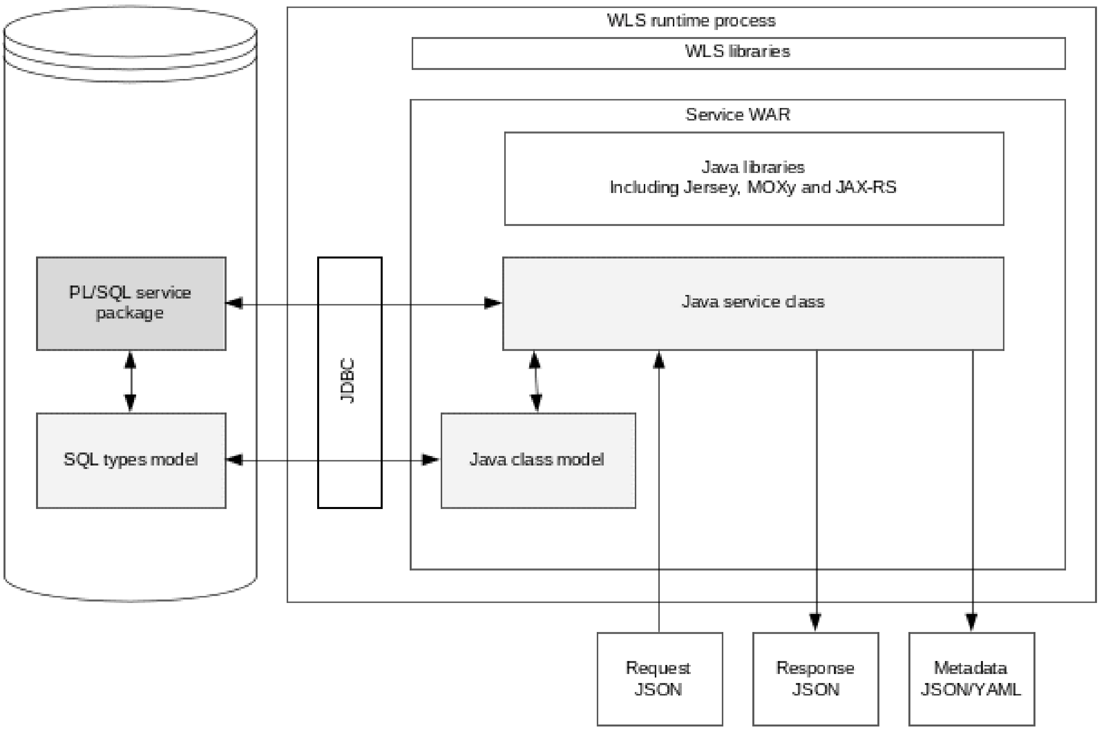
- OHIJET – The JET based web application

- PSL_AUN – Web service to log on to the OHI Back Office database using the credentials entered by the user. If successful, return an OAUTH2 access token and a list of 'claims', i.e. a list of modules which may be accessed by the user.
- PSL_JUP – Web service to retrieve the OHI Back Office parameter values for 'JavaScript user interface' as a list of properties.
- PSL_<app> – Many private REST services used by the JET base web application

The web application calls PSL web services to interact with the OHI Back Office database. Ensure that the OHIJET application is deployed successfully. The corresponding EAR file (OHIJET.ear) can be found in $OZG_BASE/java.

2.2 HSL and PSL services

The architecture of each of the PSL services is shown below:



For more information about HSL services, see **Doc[1]**.

PSL services are created specifically to support OHI BO application components like the OHIJET user interface application. 'PSL' stands for 'Private Service Layer'. These services use the same technology as the HSL services but cannot be used as an 'API'.

Characteristics of PSL services:

- Specifically built to support OHI BO applications components. This means that PSL services must not be called by customer applications. It also means that definitions, contents, or operation of PSL service operations may be changed by OHI Back Office Development without notice.
- No online help documentation.
- Built on the same technology as HSL services.
- Configured through a 'psl.properties' file, similar to the 'hsl.properties' file used for HSL services.

3 Prerequisites

The following prerequisites apply before you can deploy the OHI Back Office JET Application.

3.1 WebLogic Server (WLS) Preparation

Follow the instructions in **Doc[1]** to prepare WebLogic Server.

Ensure that the following steps are completed:

- Create PSL properties file (as described below)
- Add a reference to the `psl.properties file as a Server Start argument. See paragraph 3.3.1.`

Note: For OHIJET.ear you may use the same WLS domain as for the HSL and SVL services. The OHI Back Office JET application no longer uses any HSL services, so you may deploy the HSL services in the same or a different environment with a different authentication setup.

3.2 Database Preparation

Install the database components for OHI Back Office using OHIPATCH.pl.

3.2.1 Create a PSL database user account

The PSL services require a PSL database account similar to the HSL database account:

1. Create a database account, for example PSL_USER.
2. Grant create session system privilege to the PSL database account.
3. Log on as the OHI Back Office schema owner, enable serveroutput and run:

```
alg_security_pck.psl_grants
( pi_owner   => '<ohibo_owner>'
, pi_grantee => '<psl_user_account>'
);
```

Example:

```
execute alg_security_pck.psl_grants
( pi_owner   => 'OZG_OWNER'
, pi_grantee => 'PSL_USER'
);
```

The notes mentioned in **Doc[1]** in the section ‘Creating a HSL database account’ also apply to the PSL database account.

NOTE: do not use the same database account for PSL_USER as you used for HSL_USER. Both accounts have their own specific set of privileges that are checked and corrected by the OHI Patch procedure. Any other privileges will be revoked, so only one set will survive if you use the same account for both.

3.2.2 Create WLS data source for PSL database account

Create a data source for PSL services, similar to ‘Creating a data source’ for the HSL services (as described in **Doc[1]**). But then refer to the PSL database account instead of the HSL database account. Also be sure to use the ‘psl_grants’ packaged procedure instead of the ‘hsl_grants’ version while following the steps outlined in **Doc[1]**.

3.3 OHI Back Office

Do not forget to set up module authorization for the OHIJET modules. OHIJET uses the same database accounts for the end users and the same employee ("functionaris") registrations as the OHI Back Office Forms application.

To access module authorization: Systeem > Beheer > Autorisatie > Moduleautorisatie.

Your organization needs to set up a role structure, assign modules (OHIJET screens) to the roles, and assign role(s) to the employees.

3.3.1 Configure psl.properties

The properties controlling all PSL services are set in the psl.properties file.

With the OHI Back Office release installation, a properties file template called psl.properties.template is distributed to the $OZG_BASE/conf/Back-Office directory. Each OHI Back Office release may overwrite this template with an updated version. The psl.properties.template can be used as an example to create your own psl.properties file (for example in $OZG_BASE/conf).

Please note that all values are examples. You should consider if these values are appropriate for your installation and replace them with your own values if needed. Values indicated with `<<some_name>>` in the templates are placeholders and must be replaced. This notation is intended to make scripted deployment easier. Also make sure not to set log level to FINE, FINER or FINEST in production mode, use SEVERE or WARNING instead.

Do not set psl.developermode=true in environments with production data, because Private Health Information may be written to log files with that setting.

The PSL services use the same generic properties as the HSL services (but prefixed with `psl` instead of `hsl`). These generic properties are described in more detail in chapter 'Configuration Files for HSL services' in **Doc[1]**. In addition to the generic properties there are several properties that are specific to PSL. These will be outlined over the next paragraphs as they relate to a Single sign-On (SSO) setup.

The PSL services do support specifying different properties per service, using the <app> abbreviation. This is intended for specific circumstances only. When there is a problem, OHI Support may ask you to specify for example an enhanced log level and/or separate specific log file name for a specific PSL service. The abbreviation will be provided by OHI Support. Valid abbreviations are not published.

OHI advises you to use the PSL services with OAuth 2.0 (a 'Bearer' HTTP Authorization header with a JWT token) as a minimum authorization method, which is enforced by setting the default `psl.authorization` property to TOKEN.

3.3.2 Built-in authentication with PSL_AUN

The `psl.<app>.authorization` property selects which authorization method is allowed for a PSL service. This value defaults to the value of `psl.authorization`.

`To use the built-in OAuth 2.0 implementation for the OHIJET application and the PSL services psl.authorization` must be set to TOKEN

Allowed values:

- TOKEN – use 'Bearer' HTTP Authorization header with JWT token
- NONE – Don't use any form of authorization.
- Or a combination, where specific services might have different forms of authentication.

Note that a value of NONE is mandatory for the PSL_AUN and PSL_JUP services. The OHIJET application needs to be able to invoke these services before a user is logged in to the application. The PSL_AUN (AUtheNtication) service serves as a OAuth2 token endpoint and in addition provides information to the client as to the whereabouts of an identity provider, should a SSO setup be configured. Details regarding the JUP-service can be found in appendix G.

So, with regards to the built-in authentication, the psl.properties file should at the very least contain the following properties:

```
psl.authorization=TOKEN
psl.aun.authorization=NONE
psl.jup.authorization=NONE
psl.usercontext=<CUSTOMER SPECIFIC USERNAME>
psl.aun.usercontext.control=PROPERTY
psl.jup.usercontext.control=PROPERTY
```

Note that the usercontext properties are required because the user has no usercontext tied to their session before logging in to the application. Calls to the PSL_AUN and PSL_JUP services before a user has logged in will be handled (and for example shown in log files) with a usercontext referring to the value given to `psl.usercontext`.

The <CUSTOMER SPECIFIC USERNAME> should be replaced with a generic Back Office employee account ('functionaris'). Additionally:

- This employee account should NOT be linked to the PSL_USER database account used in the Data Source.
- The Back Office employee account should not have a linked database account.

3.3.3 Authentication with Oracle Access Manager

The OHIJET application can be configured to enable authentication with Oracle Access Manager. To achieve this some properties must be set in psl.properties.

- `psl.sso.platform` – The platform to be used for SSO. Set this to 'oam'. Allowed values: oam, openid, none
- `psl.sso.session` – The endpoint to validate session_id in OAM, for example: https://www.myoamserver.nl:1234/oam/services/rest/access/api/v1/session/{id}
- `psl.sso.authentication` – The basic authentication header for OAM REST API calls. This is a base64 encoded string of the WebLogic username:password for OAM. Note that the value of this property will be encrypted upon deployment.

3.3.4 Authentication with OpenID Connect

The OHIJET application can be configured to enable authentication with an OpenID Connect based Identity Provider such as Oracle Identity Cloud Service (IDCS). The required setup has been described in detail in the whitepaper published to the OHI Back Office community of technical administrators.

Several properties in the psl.properties file relate to the setup of SSO with the OHIJET application:

- `psl.sso.platform` – The platform to be used for SSO. Set this to 'openid'. Allowed values: oam, openid, none
- `psl.sso.openid.authorizationendpoint` - Endpoint to generate Authorization Code and Identity Token.
  Example: https://idcs.oc9qadev.com/oauth2/v1/authorize

- psl.sso.openid.tokenendpoint - Endpoint to generate Access Token. Example: https://idcs.oc9qadev.com/oauth2/v1/token
- psl.sso.openid.endsessionendpoint - Endpoint to logout the user and remove all associated cookies and sessions.
  Example: https://idcs.oc9qadev.com/oauth2/v1/userlogout
- psl.sso.openid.clientid - The unique client ID for the OHIJET application on the authorization server.
- psl.sso.openid.clientsecret - The secret provided by the authorization server and known by OHIJET. Note that the value of this property will be encrypted upon deployment. Make sure to set the property outlined in chapter 3.3.5.

3.3.5 Signature Encryption

The encryption algorithm used by PSL services is driven by the PSL property psl.tokenvalidation.rotor=*your_secret_key*

Keep the value of psl.tokenvalidation.rotor secret and limit the access to the properties file at the OS level.

3.3.6 Setting user context

Every operation of a REST service must be executed by an OHI employee account (Dutch: functionaris). This is a registered user of the OHI BackOffice application.

The default account is set through psl.<app>.usercontext (defaults to psl.usercontext).

If token validation is used, the usercontext is retrieved from the JWT access token.

The following configuration parameters are used:

- psl.<app>.usercontext - defaults to the value of psl.usercontext
  The user context which must be used for executing an operation.
  The value must be a registered OHI employee (in Dutch: 'functionaris').

- psl.<app>.usercontext.control - defaults to the value of psl.usercontext.control
  Allowed values:

  a. PROPERTY
     Use the value of psl.<app>.usercontext to set the user context.

  b. TOKEN
     Retrieve the user context from the access token.
     Note that psl.<app>.usercontext.claim **must** be set to indicate which field in the JWT access token contains the usercontext.

If psl.<app>.usercontext.control is set to TOKEN, the following configuration parameters control how the usercontext is retrieved:

- psl.<app>.usercontext.token.type - defaults to the value of psl.usercontext.token.type
  Set the type of token.
  Allowed values: JWT

- psl.<app>.usercontext.claim - defaults to the value of psl.usercontext.claim
  Determines which field in the JWT token contains the usercontext. Claim is a standard JSON object of a JTW token. In the example below prn is a reserved claim name for the principal, the subject of the JWT.
  Example: psl.<app>.usercontext.claim=prn

3.3.7 WLS Managed Server Start arguments for psl.properties

Before using the web services, implement the following actions as described below. These actions must be executed only once. There is no need to repeat them when you update a deployment or delete and install it again.

Add a Server Start argument by adding a line to the file $DOMAIN_HOME/bin/setUserOverrides.sh (See **Doc[1].** Add the line to the part for the PSL server, as indicated below:

```
# specify additional java command line options for specific servers
if [ "${SERVER_NAME}" = "ms_ohi_psl" ]
then
# Set location for PSL properties file
JAVA_OPTIONS="-Dpsl.properties="/u01/domains/OHIDEV01/conf/psl.properties" ${JAVA_OPTIONS}"
fi
export JAVA_OPTIONS
```

- Make sure to keep the parts with ${JAVA_OPTIONS} on the same line.

This example uses a properties file with the name psl.properties which is located in the $OZG_BASE/conf folder of your OHI Back Office application server environment, but you can specify any name and location.

When completed, (re)start the Managed Server. This can be done from the WebLogic Admin console, or from the command line with the following commands:

```
cd $DOMAIN_HOME/bin
./startManagedWebLogic.sh ms_ohi_psl http://localhost:7061
```

The example above contains the Managed Server’s name as first parameter and the listen address of the Admin Server of the domain as second parameter.

Check in the <ManagedServer>.log file and/or in the <ManagedServer>.out file in $DOMAIN_HOME/servers/<ManagedServer>/logs directory of your Managed Server whether the command line contains the arguments as specified above.

3.4 Deployment

After the WebLogic setup is completed and the properties files are configured, the web services can be deployed. The procedure for deploying the OHIJET application is described in chapter 4 “Installation of OHIJET Application“.

4 Installation of OHIJET Application

The OHI Back Office web modules and REST services are packaged in a single archive named 'OHIJET.ear'. This ear file must be deployed to WLS.

Until OHI Back Office release 10.19.1.3 all WAR files related to OHIJET were deployed separately. Before deploying OHIJET.ear ensure that none of the following WAR files are deployed:

1. PSL_<app>.war: Ensure that no PSL services are deployed. If PSL services do exist they need to be deleted in the WLS console.
2. ZRGOHIJET.war: Ensure that this WAR file is not deployed. If this file does exist it needs to be deleted in the WLS console.

4.1 OHIJET Deployment

Although the procedure is similar to deploying HSL services you may find it useful to look at the step-by-step instructions below.

- Select 'Deployments' from the WLS console:
- Select 'Install' to create a new deployment.
- Select 'OHIJET.ear'
- Select to install the deployment as an application
- Target the application at the managed server on which the OHIJET application should become available.
- Finish the deployment using 'Custom Roles and Policies'.

The application is now deployed.

4.2 Post Installation

New deployments are not initially active. Either start the new deployment through the WLS console or restart the managed server(s) to which the application is deployed.

NOTE: If the file specified by psl.properties cannot be read, messages as below will show up in the WebLogic console and in the <ManagedServer>.log file:

```
java.lang.RuntimeException: Property file could not be loaded.
….
java.io.FileNotFoundException:
/u01/domains/OHIDEV01/conf/psl.properties (No such file or directory)
```

4.3 Updating the OHIJET Application

In time, more modules will be added to the OHIJET application.

When deploying a newer version of the OHIJET application, select 'Update' to update the application (ensure that the new OHIJET.ear is copied to the existing location).

Alternatively, select 'Delete' to delete the existing OHIJET application and create a new deployment as described above.

4.4 Deployment validation

When validating the deployment, all components must be correctly installed and deployed.

Before you continue:

- verify that you have completed all steps in 'Prerequisites'.
- verify that you have deployed and started the OHIJET.ear application as described in this chapter.

4.4.1 Use a certified browser

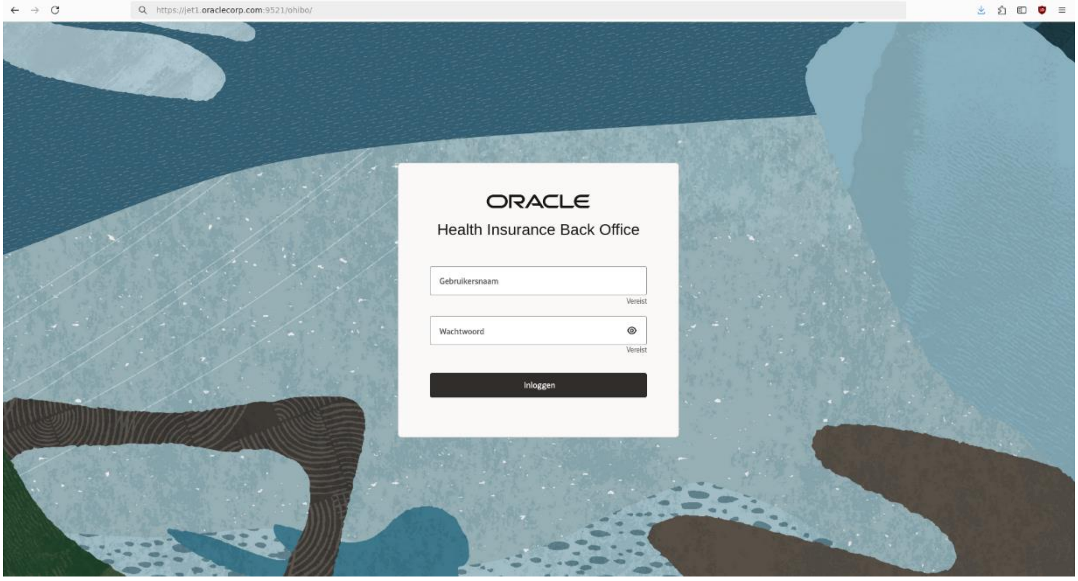
The JET application must be run using a certified browser. To find out which versions of which browsers are certified for an OHI Back Office release, see the document "*OHI_BO_certifications*" on Secure Sites. The section "OHIJET" in paragraph "Browser support" has a list of supported browsers and the minimum versions that are certified.

4.4.2 Run the JET application

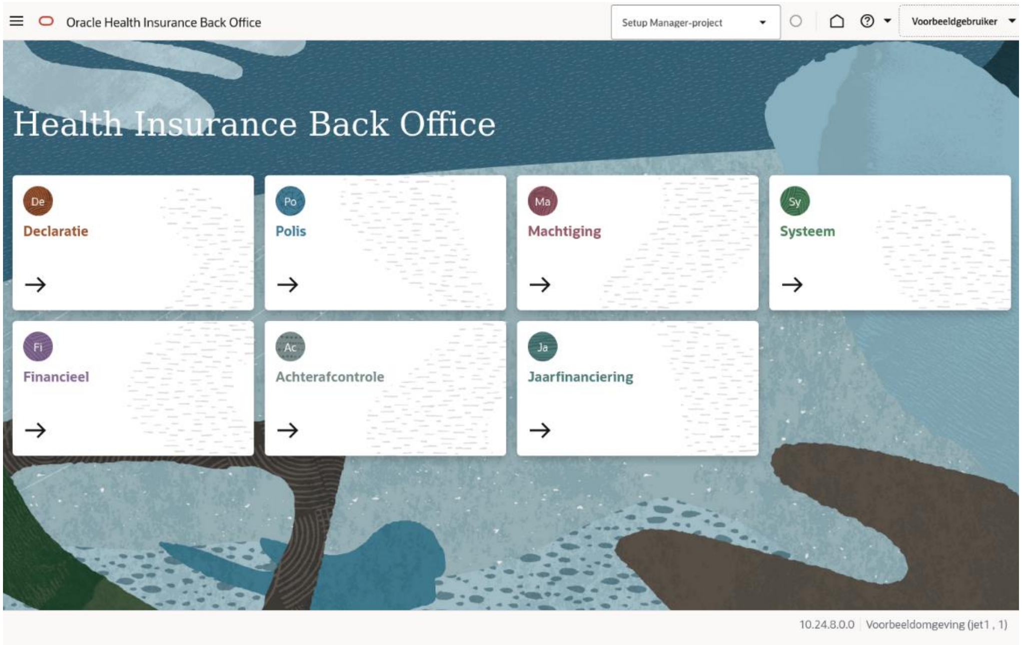
To verify the deployment, browse

`https://<myserver.myorganisation.com>:<myport>/ohibo`

Example:


https://jet1.oraclecorp.com:9521/ohibo/
ORACLE
Health Insurance Back Office
Gebruikersnaam
Vereist
Wachtwoord
Vereist
Inloggen

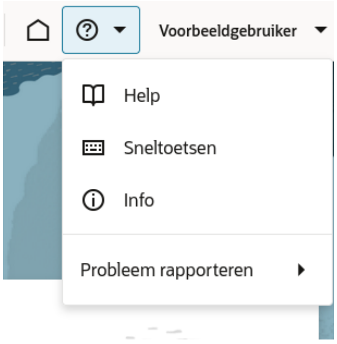
When you have successfully logged on with your OHI username and password, the screen should look like this:

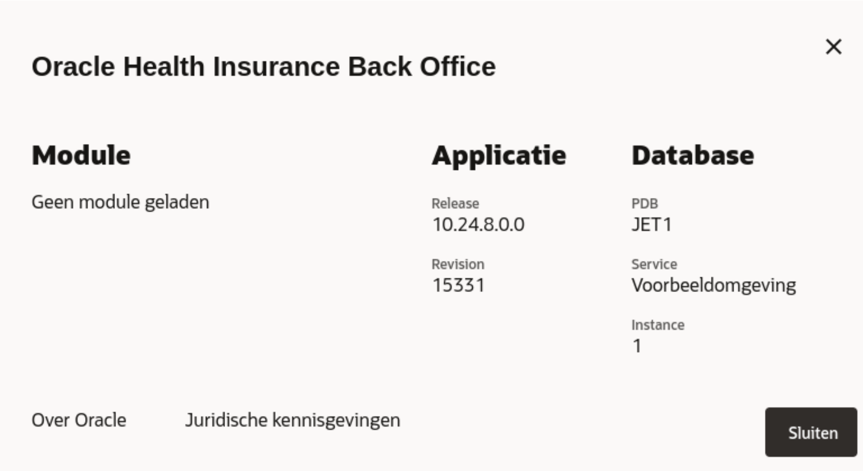


NOTE: this requires authorization for at least one OHIJET Module.

At the bottom right you can see the release number, the name of the database, and if a value was specified for Back Office parameter 'Naam van de omgeving', additionally you will also see that name at the bottom of the page.

Further general information relating to the JET application can be found in the 'Info' screen, which can be accessed through the drop-down menu of the question mark-icon in the upper right corner.





Oracle Health Insurance Back Office

Module

Geen module geladen

Applicatie

Release
10.24.8.0.0

Revision
15331

Database

PDB
JET1

Service
Voorbeeldomgeving

Instance
1

5 Appendix A – PSL_JUP Service

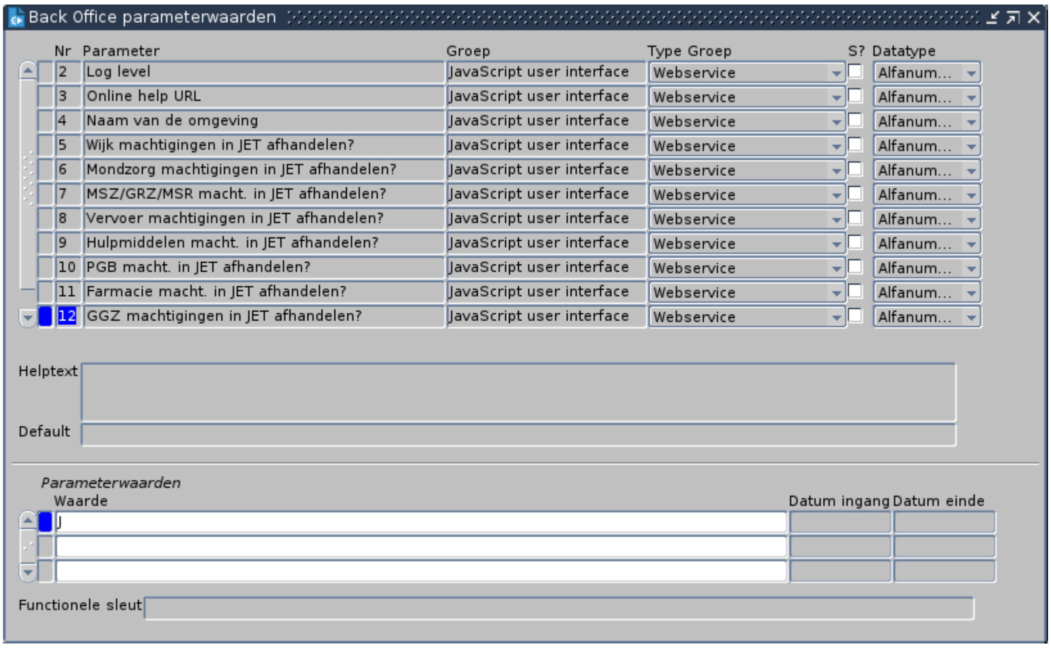
The OHIJET application uses the PSL_AUN module for authentication of its users. It uses the other PSL services as a backend to access OHI Back Office data.

Before it can do so, it needs to locate a Base URL where these services (all services that are used, so PSL_AUN and the PSL services) can be found.
This is where PSL_JUP comes in. This service, running in the same managed server as OHIJET itself, will connect to the OHI Back Office database and retrieve the OHI Back Office parameters of group 'JavaScript user interface' to retrieve the Online help URL and log level.

5.1 Back Office parameters

To set the Back Office parameters:

- Open the Forms Application and select 'Systeem > Beheer > Algemeen > Back Office parameter waarden'.
- Select group 'JavaScript user interface' and perform Execute Query.
- Optionally, set the log level, the Online Help URL, the name of the environment (displayed on the OHIJET start screen)



Back Office parameterwaarden

Nr	Parameter	Groep	Type Groep	S?	Datatype
2	Log level	JavaScript user interface	Webservice	☐	Alfanum...
3	Online help URL	JavaScript user interface	Webservice	☐	Alfanum...
4	Naam van de omgeving	JavaScript user interface	Webservice	☐	Alfanum...
5	Wijk machtigingen in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
6	Mondzorg machtigingen in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
7	MSZ/GRZ/MSR macht. in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
8	Vervoer machtigingen in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
9	Hulpmiddelen macht. in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
10	PGB macht. in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
11	Farmacie macht. in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...
12	GGZ machtigingen in JET afhandelen?	JavaScript user interface	Webservice	☐	Alfanum...

Helptext

Default

Parameterwaarden

Waarde	Datum ingang	Datum einde
J		

Functionele sleut

5.2 PSL Properties

In addition to the usual PSL properties (see 'Back Office PSL properties file' above), the PSL property `psl.jup.authorization` in psl.properties should be set to a value that differs from the value for the other PSL services. The value must be set to `NONE`. The reason is that anyone must be able to call PSL_JUP to retrieve the OHI BO parameters for the JavaScript UI of OHIJET.

The data source (`psl.jup.jndiname`) is used to retrieve Back Office Parameters for the OHIJET Application.

☞ WARNING: for production environments, set the psl.jup.loglevel to NONE or leave it empty, to avoid logging of sensitive data.