Generating Reports in WAY4

Table of Contents

Introduction

This document is intended for WAY4™ system administrators (bank or processing centre employees), responsible for Reports Engine R2 setup. Reports Engine R2 is used to generate reports.

When working with this document, it is recommended to use the following resources from the OpenWay documentation series:

- "Scheduler R2".

The following conventions are used throughout the document:

- Field labels in screen forms are shown in *italics*.
- Screen form button labels are encased in square brackets, such as [Approve].
- Sequences for selecting user menu items are given with arrows, as in Issuing → Contracts Input & Update.
- Sequences for selecting system menu items, are given with a different type of arrow, for example Database => Change password.
- Key combinations used when working with DB Manager are shown in angular brackets as in <Ctrl>+<F3>.
- Values that vary for each local instance of the program, such as the names of directories and files, as well as file paths are displayed in angular brackets, such as <OWS_HOME>.

Warnings about potentially hazardous situations or actions.

Information about important features, additional options or the best use of certain system functions.

Chapter 1. Purpose

Reports Engine R2 is used to generate reports.

In Reports Engine R2 a single setup mechanism is used for WAY4 client applications (DB Manager, WAY4 Manager, WAY4 Remote Access, WAY4Web Client, Scheduler R2).

Report generation modes:

- Batch mode.
- Single reports.

Chapter 2. Installing Oracle Reports Services

Oracle Reports Server installation

Introduction

This document describes OS pre-configuration and the installation/configuration process for Oracle Reports Services for use with OpenWay products. OpenWay software installation and configuration processes are not described here.

Complete documentation can be found in the Oracle Help Center at https://docs.oracle.com /middleware/12212/formsandreports/install-fnr/toc.htm

Choosing the version to install

The main Oracle component containing Oracle Reports Services 12.2.1.2 is Oracle Fusion Middleware.

Note: it is possible to upgrade an already installed Oracle Reports Server to version 12.2.1.2. Follow these official instructions: https://docs.oracle.com/middleware/12212/formsandreports /install-fnr/GUID-20AC6348-AA11-4D36-A460-8C06A0218A66.htm#FRINS420

This document describes installation of version 12.2.1.2 with a graphical user interface.

Choosing the operating system for the installation

Check the compatibility matrix provided by Oracle and select the compatible OS. This information can be found at http://www.oracle.com/technetwork/middleware/ias/downloads /fusion-certification-100350.html

Some systems may have different requirements for pre-configuration. Pre-configuration for Red Hat Enterprise Linux 7 is described here.

Please follow official Oracle Fusion Middleware documentation for calculating server hardware requirements: http://docs.oracle.com/html/E77908_01/toc.htm

Obtaining required software

For 12.2.1.2 Reports, the following packages must be obtained from Oracle:

1. Fusion Middleware Infrastructure v. 12.2.1.2.0
2. Fusion Middleware Forms and Reports v. 12.2.1.2.0

For Fusion Middleware 12.2.1.2, a sys database connection to Oracle Database server is required. Check the System Requirements and Supported Platforms for Oracle Fusion Middleware document. It is recommended to use Oracle DB 12c with all the latest DB patches. A remote Oracle DB server can be used in the infrastructure or the Oracle DB can be installed directly on the server that is used for running Oracle Fusion Middleware Infrastructure with Oracle Fusion Middleware Forms and Reports. In the latter case, hardware requirements for the server must be extended according Oracle DB 12c documentation.

OS pre-configuration

Ensure that the system has all necessary security patches before installing Oracle Reports Server.

The following shows how to pre-configure Red Hat Enterprise Linux 7 with default packages and GUI. Please use the official documentation for the selected OS to perform the following steps.

1. Prepare the server. Register the software subscription and apply all latest patches (RHEL Network for RHEL 7 OS). Disable the firewall and SELinux if they are not necessary, or configure them using the standard documentation. If the firewall is enabled, this may block running the Reports Services.
2. Create an `oinstall group and oracle` user. Include the `oracle` user in the group that was created.
3. Disable ipv6 if it's not required (https://access.redhat.com/solutions/8709 for RHEL 7 OS).
4. Create the directory `/oracle/middleware` and allow the `oracle:oinstall` user write permissions for this directory.
5. Copy Fusion Middleware Infrastructure and Fusion Middleware Forms and Reports packages to the destination server in Oracle user's environment and make the `*.bin` files executable.
6. Both installers may be run in the command line interface, but installation using a graphical user interface is described here. Install the necessary graphical environment on the server or configure configure application display forwarding (X11 forwarding) to the remote X server over XDMCP or SSH . For a minimal installation (headless) of RHEL 7 OS and a remote connection for installing Oracle Report Services, the following packages must be installed: `xauth, libXtst compat-libcap1, libstdc++-devel, sysstat, gcc, gcc-c++, ksh, glibc-devel, libaio-devel, motif`.
7. Also, for RHEL 7 OS, it is necessary to enable the `rhel-7-server-optional-rpms` repository (edit `/etc/yum.repos.d/redhat.repo` manually or use the command `subscription-manager repos --enable=rhel-7-server-optional-rpms`) and then additionally install the `compat-libstdc++-33` package. Next, create a symbolic link from `/lib64/libXm.so.4` to `/lib64/libXm.so.3`
8. Download and install the latest Oracle JDK 1.8 version. Set up the `JAVA_HOME` environment variable and add the `$JAVA_HOME/bin` directory to PATH.
9. Connect to the server via the graphical user interface or via SSH with the X forwarding option. If the XDMCP protocol is used, check that the `DISPLAY` environment variable points to the remote X server and this server allows remote connections.

Installation

1. All installation and configuration steps should be performed using the local account `oracle:oinstall` unless otherwise indicated.
2. As the `oracle` user, run the `java -jar fmw_*_infrastructure.jar` command to start the Fusion Middleware Infrastructure installer.
3. Set Oracle Home to `/oracle/middleware`. Select the type of installation "`Fusion Middleware Infrastructure`". Set Inventory Directory to `/home/oracle/oraInventory` and Operation System Group `oinstall`.

4. Then install Oracle Fusion Middleware Forms and Reports. Run the installer by executing the `.bin` package. Follow the wizard and select the same `ORACLE_HOME that` was used for installing FMW Infrastructure.
5. In the fourth step of the wizard step, select `Forms and Reports Deployment`.
6. When Oracle FMW Forms and Reports installation has been completed, prepare the database schemas which are necessary to install and run Oracle Reports. Run the `rcu` script from the `$ORACLE_HOME/oracle_common/bin` directory. Use the option `Create Repository` and the `System Load and Product Load`item. Provide the `sys` database connection, change the schema prefix (for example, to `FMW`) and select the schemas `FMW_STB, FMW_OPSS, FMW_IAU, FMW_IAU_APPEND, FMW_IAU_VIEWER`. Provide passwords and tablespaces for them (if tablespaces do not exist, default tablespaces will be created).
7. Run the `config.sh` script from `$ORACLE_HOME/oracle_common/common/bin`.
8. Create a new domain (`/oracle/middleware/user_projects/domains/base_domain`) `base_domain` and select the following components in the next step: `Oracle Reports Server, Oracle Reports Application, Oracle Reports Tools, Oracle Enterprise Manager, Oracle HTTP Server (Collocated)`. This directory will be your `DOMAIN_HOME`.
9. Set Domain Mode to `Production`. *Note*: in this case, no `boot.properties` file will be generated by default, so domain startup should be in interactive mode.
10. In the sixth step of the wizard, provide the `FMW_STB` database schema connection and specify the RCU autoconfiguration option. Click the `Get RCU Configuration` button and after checking, go to the next step.
11. Perform Advanced Configuration for all steps and review the configuration.

Sample configuration

- Administration Server: `AdminServer`: Listen external IP address, port `7001`, Server Groups: `JRF-WS-CORE-MAN-SVR`
- Node Manager: `Per Domain Default Location`, provide `Node Manager Credentials`
- Manager Servers: `WLS_REPORTS`: Listen external IP address, port `9002`, Server Groups: `REPORTS-APP-SERVERS`
- Clusters: `cluster_reports`: Dynamic Server Groups: `WSM-CACHE-DYN-CLUSTER`
- Server Templates: `wsm-cache-server-template:` Listen Port `7100`, Cluster: `cluster_reports`
- Dynamic Servers: `cluster_reports, wsm-cache-serverd, wsm-cache-server-template`. `Calculated Machine Names` checked, `Calculated Listen Ports` checked, `Dynamic Cluster` checked
- Assign Servers to Clusters: `WLS_REPORTS` is assigned to `clusted_reports`
- Coherence Clusters: `default`
- Machines: `AdminServerMachine`, bind to Node Manager `localhost:5556`, no Unix Machines

- Assign Servers to Machines: `AdminServer` and `WLS_REPORTS` are assigned to AdminServerMachine
- Virtual Targets: no
- Partitions: no
- System Components: no
- Domain Frontend Host: unchecked (do not configure)
- Deployment Targeting: no changes
- Services Targeting: no changes
- Configuration Summary: review and confirm (`Create`)

Post-configuration

1. Configure environment variables for the oracle user

```
export ORACLE_HOME=/oracle/middleware
export DOMAIN_HOME=$ORACLE_HOME/user_projects/domains/base_domain
export PATH=$ORACLE_HOME/bin:$PATH
```

2. Start `AdminServer` and `WLS_REPORTS`. When the first deployment has been completed, all configuration files will be ready for editing.
3. Run `$DOMAIN_HOME/startWebLogic.sh`. When asked, provide the password.
4. When WebLogic is started, open http://<your_host>:7001/console and log in.
5. To run `WLS_REPORTS` and `OHS` servers from the WebLogic console, Node Manager must be running. Open `$DOMAIN_HOME/bin` and run script`startNodeManager.sh`
6. Create Reports Components. Run the `wlst` utility from `$ORACLE_HOME/oracle_common/common/bin/wlst.sh` and then run the following commands:

```
connect("weblogic","password", "<your_host>:7001")
createReportsToolsInstance(instanceName='reportsToolsInstance', machine='AdminServerMachine')
createReportsServerInstance(instanceName='reportsServerInstance', machine='AdminServerMachine')
exit()
```

1. Open *base_domain->Environment->Servers->Control* in the WebLogic console and run `WLS_REPORTS`. Wait for it to start and then check that rwservlet is running: open URL http://<your_host>:9002/reports/rwservlet
2. Enable diagnostic output. This is very useful during configuration of the server. This feature can be disabled later, if necessary.

Edit `$DOMAIN_HOME/config/fmwconfig/servers/WLS_REPORTS/applications/reports_12.2.1/configuration/rwservlet.properties` and add the following parameter below the `<inprocess>` parameter:

```
<webcommandaccess>L2</webcommandaccess>
```

> `WLS_REPORTS` should be restarted for applying changes

3. Default security settings for Reports Server require user authentication and authorization to access `rwservlet` . Configure users with Oracle Enterprise Manager for internal users and roles according to the official manual (or connect the Reports Server with Oracle Access Manager and Oracle Internet Directory) and authenticate requests or disable authentication. See https://docs.oracle.com/middleware/1221/formsandreports/use-reports/pbr_sec_arch.htm#RSPUB0015 for details.
4. To disable authentication and authorization, open `$DOMAIN_HOME/config/fmwconfig/servers/WLS_REPORTS/applications/reports_12.2.1/configuration/rwserver.conf` and do as follows:

   4.1 Find the `<security>` tag with id `rwJaznSec` and defined class `oracle.reports.server.RWJAZNSecurity` and comment it

   4.2 Find the `<job>` tag with defined attribute *securityId* `rwJaznSec` and remove this attribute with its value

   4.3 Restart `WLS_REPORTS` server to apply changes

Configuring init scripts

Oracle Reports server can be started when the OS starts up. Note that WLS_REPORTS or/and AdminServer should be configured with the boot.properties file for authentication and authorization on startup. See Item 9 in the Installation above. The following tutorial provided by Oracle can also be used: http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/wls/12c/15-BootProp--4471/bootproperties.htm

In fact, this is a simple operation: if the server must be started on system startup, create a security subdirectory in $DOMAIN_HOME/servers/<Server Name> after installation and add a new boot.properties text file with the following content:

```
username=<weblogic username>
password=<weblogic password>
```

Use clear passwords. The first time the server starts in non-interactive mode, both parameters will be encoded.

RHEL 7 OS used in the sample installation does not have init.d daemon because it was replaced by system.d daemon. Use the operating system's specific startup scripts for starting the necessary services.

1. Create a startup script for `WLS_REPORTS` and `OHS` to start them via Node Manager.

   Create the file `/oracle/middleware/user_projects/domains/base_domain/runServer.py` with the following content (replace the values, when necessary)

```
nmConnect('<weblogic user>','<weblogic password>','localhost','5556','base_doma
in','/oracle/middleware/user_projects/domains/base_domain','SSL')
nmStart(serverName='<ohs server name>',serverType='OHS')
nmStart('WLS_REPORTS')
nmStart('AdminServer')
exit()
```

2. Create a shutdown script for `WLS_REPORTS` and `OHS` to stop them via Node Manager.

   Create the file `/oracle/middleware/user_projects/domains/base_domain/stopServer.py` with the following content (place your values where it is necessary)

```
nmConnect('<weblogic user>','<weblogic password>','localhost','5556','base_doma
in','/oracle/middleware/user_projects/domains/base_domain','SSL')
nmKill(serverName='<ohs server name>',serverType='OHS')
nmKill('WLS_REPORTS')
nmKill('AdminServer')
exit()
```

3. As the root user, open the directory /etc/systemd/system
4. Here, create startup scripts for Node Manager and Oracle Report Server. If Oracle DB for Oracle Fusion Middleware was installed locally, also add the init script for Oracle DB and create a dependency between it and Fusion Middleware Servers. Do not forget to enable DB startup in /etc/oratab also.

oracledb.service

```
# /etc/systemd/system/oracledb.service
# Invoking Oracle scripts to start/shutdown Instances defined in /etc/oratab
# and starts Listener
[Unit]
Description=Oracle Database(s) and Listener
Requires=network.target

[Service]
Type=forking
Restart=no
```

```
ExecStart=/oracle/database/product/12.1.0/dbhome_1/bin/dbstart /oracle/database
/product/12.1.0/dbhome_1
ExecStop=/oracle/database/product/12.1.0/dbhome_1/bin/dbshut /oracle/database
/product/12.1.0/dbhome_1
User=oracle

[Install]
WantedBy=multi-user.target
```

nodemanager.service

```
[Unit]
Description=NodeManager service
Wants=oracledb.target
After=oracledb.target

[Service]
Type=forking
WorkingDirectory=/oracle/middleware/user_projects/domains/base_domain
ExecStart=/bin/sh -c "/oracle/middleware/user_projects/domains/base_domain/bin
/startNodeManager.sh 2>&1 | tee /tmp/nodemanager.log &"
ExecStop=/oracle/middleware/user_projects/domains/base_domain/bin
/stopNodeManager.sh
ExecStartPost=/bin/sh -c "while ! grep -q 'started on port 5556' /tmp
/nodemanager.log; do echo 'Starting...'; sleep 10; done;"
TimeoutSec=320
User=oracle

[Install]
WantedBy=multi-user.target
```

reportsserver.service

```
[Unit]
Description=Admin Server and Managed Server WLS_REPORTS with Oracle HTTP Server
After=nodemanager.service
Requires=nodemanager.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/oracle/middleware/wlserver/common/bin
ExecStart=/bin/sh -c "/oracle/middleware/oracle_common/common/bin/wlst.sh
/oracle/middleware/user_projects/domains/base_domain/runServer.py 2>&1 | tee
/tmp/reportsserver_start.log"
```

```
ExecStartPost=/bin/sh -c "while ! grep -q 'Successfully started server
AdminServer' /tmp/reportsserver_start.log; do echo 'Starting...'; sleep 20;
done;"
ExecStop=/bin/sh -c "/oracle/middleware/oracle_common/common/bin/wlst.sh
/oracle/middleware/user_projects/domains/base_domain/stopServer.py 2>&1 | tee
/tmp/reportsserver_stop.log"
User=oracle

[Install]
WantedBy=multi-user.target
```

5. Execute the `systemctl daemon-reload` command
6. Enable configured services

```
systemctl enable reportsserver
systemctl enable nodemanager
systemctl enable oracledb
```

7. Now the following commands can be used:

```
systemctl status reportsserver # for getting the status of service
systemctl start reportsserver # for starting the service
systemctl stop reportsserver # for shutting down the service
systemctl list-units # for getting the complete services configure at the
server
```

Configuring Reports Server for OpenWay products

1. Set up the character set for decoding URL parameters. AMERICAN_CIS.CL8MSWIN1251 is recommended for Cyrillic, UTF8 is also possible.

   Open `$DOMAIN_HOME/reports/bin/reports.sh`, find the following construction

```
if [ "$NLS_LANG" = "" ]
then
    NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
fi
```

   and add the string

```
NLS_LANG=AMERICAN_CIS.CL8MSWIN1251; export NLS_LANG
```

before it.

2. Configure the maximum engines for in-process Reports server using the configuration file `$DOMAIN_HOME/config/fmwconfig/servers/WLS_REPORTS/applications/reports_12.2.1/configuration/rwserver.conf` (see the `<engine>` tag with id `rwEng`). See the official documentation at https://docs.oracle.com/middleware/1221/formsandreports/use-reports/pbr_arch003.htm#RSPUB23215 for other options (timeout, queue size, cache directory location and its size etc).
3. Set up fonts for Oracle Reports. When using non-Latin (Cyrillic for instance) character sets, Oracle Reports font settings should also be changed. If this is not done, garbage characters may appear in the generated reports:

   3.1 In the files from the `$ORACLE_HOME/guicommon/tk/admin/AFM` folder, replace the `AdobeStandardEncoding` value with `FontSpecific`. It is sometimes sufficient to change "Courier" files only.

   3.2 For PDF-output, it is recommended to:

   3.2.1 Install required ttf fonts (copy *.ttf files into the `$DOMAIN_HOME/reports/fonts` folder. Ensure Oracle Reports supports the font format supplied).

   3.2.2 Set mapping for copied fonts in the `$DOMAIN_HOME/config/fmwconfig/components/ReportsToolsComponent/reportsToolsInstance/guicommon/tk/admin/uifont.ali` file in the [PDF:Subset] section.

   For instance, if cour.ttf ("Courier New" font), couri.ttf ("Courier New" italic), etc were copied to be used instead of “Courier”, the following must be added to the [PDF:Subset] section:

```
"Courier New"..Italic.Bold.. = "courbi.ttf"
"Courier New"...Bold.. = "courbd.ttf"
"Courier New"..Italic... = "couri.ttf"
"Courier New"..... = "cour.ttf"
```

   3.3.3 Remove mapping that is already present in the uifont.ali in [Global] and [PDF] sections.

   For instance, after specifying "Courier New" in [PDF:Subset], remove "Courier New" = courier found in [Global].

The Reports Server (`WLS_REPORTS`) must be restarted for changes to take effect.

Changing HTTP/HTTPS Port Numbers and configuring HTTPS (SSL)

It is recommended that the Oracle Reports HTTP/HTTPS server ports be changed to the default values 80/443 in order to access HTTP/HTTPS services without specifying a port number.

The recommended steps are (with Oracle HTTP Server (`ohs`)):

1. Open the Enterprise Manager console at http://<your_host>:7001/em and log in.
2. Open *Administration -> OHS instances*. Lock the session and create a new OHS instance. Activate changes.
3. Using *Target Navigation* (the main navigation menu) open settings for the OHS server that was create. Open *Administration -> Virtual Hosts*

4. Lock the session and add an SSL port (`*:443`). Other fields can be skipped, just provide a port. Once it has been created, it is HTTP. Select the *string with 443* port in the table, then click the marker (to the right of the Configure button to open a drop-down list box) and select `SSL Configuration`. Select the *Enable SSL* checkbox and provide a `Server Wallet`. By default a pre-generated self-signed certificate is used. Use the `orapki` (https://docs.oracle.com/cd/B28359_01/network.111/b28530/asoappf.htm#i634246) utility to create a new wallet and import the certificate to it. Select the necessary `Cipher Suites` and finish configuration. Activate changes.
5. The application waiting for incoming connections to the standard SSL port (443) may be started by the root user; it may also be started by another user, on the condition that an `SUID` bit is set in OHS binaries.
6. Open the $ORACLE_HOME/ohs/bin directory as the `root` user. Do as follows:

```
chown root httpd apxs launch
chmod +s httpd apxs launch
```

7. Using *Target Navigation,* open settings for `WLS_REPORTS` (*base_domain -> cluster_reports -> WLS_REPORTS*). Open *Administration -> General Settings* there. Lock the session. Change Listen Address to `127.0.0.1`. Save and activate changes.
8. Open *Administration -> mod_wl_ohs Configuration*. Lock the session and add `Location /reports` on `127.0.0.1` with port `9002`. Activate changes.
9. Open the OHS configuration file `$DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs/mod_wl_ohs.conf` and add the following option to the configured Location `/reports`:

```
WLSRequest On
```

10. Also, add the same settings to `$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs/mod_wl_ohs.conf`. This is necessary because the change was not performed with Enterprise Manager.
11. Restart `WLS_REPORTS` using Enterprise Manager
12. Start the OHS server using Enterprise Manager and check that `rwserlet` is now available at the URL https://<your_host>/reports/rwservlet

Enable SSL for the WebLogic console and Enterprise manager

Use the standard SSL configuration, provided by WebLogic (see the standard documentation) or proxy requests through the standard SSL port. For the latter, do as follows:

1. Open the WebLogic console at http://<your_host>:7001/console and then open *Environment->Servers->AdminServer*. Lock the configuration and on the General tab find `Advanced` options. Then set `yes` for `WebLogic Plug-in Enabled`. Save changes, activate the configuration.

2. Open Enterprise Manager at http://<your_host>:7001/em. Open the AdminServer configuration. Configure Listen Address for AdminServer to `127.0.0.1` at *Administration -> General Settings*.
3. Open the OHS configuration. Configure mod_wl_ohs by adding new Locations:

```
/console 127.0.0.1 7001
/em 127.0.0.1 7001
```

4. Activate changes.
5. Open the OHS configuration file `$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs/mod_wl_ohs.conf` and add the following option to all configured locations (/reports, /console, /em):

```
WLSRequest On
```

6. Also, add the same settings to `$DOMAIN_HOME/config/fmwconfig/components/OHS/instances/ohs/mod_wl_ohs.conf`. It is necessary because change was not performed with Enterprise Manager. If you don't see `Location`(s) defined in this configuration file, copy them from `$DOMAIN_HOME/config/fmwconfig/components/OHS/ohs/mod_wl_ohs.conf`.
7. Restart the OHS server.
8. Restart AdminServer to apply the changes (including changes that were made in the first step). First, force shutdown from the Enterprise Manager console. Then run the script `startWebLogic.sh` from `$DOMAIN_HOME` from the unix console. Provide a login and password to start the server if startup is interactive.

> To set up a proxy for SSL protected ports, use the official documentation, which can be found at https://docs.oracle.com/middleware/1212/webtier/PLGWL/oracle.htm#PLGWL553

If SSL is configured on the server, enable SSL on each client computer

1. If it has not been done yet, import the Certification Authority certificate (with which the wallet certificate has been signed) into each client computer as a Trusted Root Certification Authority. If the certificate is signed by the root CA, it is enough to add only the root CA to the all clients.
2. Change client configurations to use HTTPS instead of HTTP to receive reports (for WAY4 DB Manager/WAY4 Manager, change the Reports Server URL in the "db.ini" file to https://<Reports Server host name with domain>:<SSL Port>/reports/rwservlet?+).

Converting reports

Oracle Reports templates are provided in RDF format. This is binary format and these files are compiled to the version of Oracle Reports installed in the OpenWay environment. To use these templates in production, they must be recompiled according to the version of Oracle Reports server installed in the production environment.

The simplest way to do this is to recompile RDF templates directly on the Oracle Reports server.

There are two options: as the "oracle" unix user used for running Oracle Reports Services or as any other user who has access to the OS shell and is included in the "oinstall" unix group (default group for the "oracle" unix user). In the second option, it may be necessary to grant additional privileges for access to Oracle Reports binary files and libraries.

Specify the database connection for the installed Oracle DB schema that will be used as the data source for each report. The rwconverter utility is used for conversion.

The following example shows how to convert a single report:

Export variables ORACLE_HOME and ORACLE_INSTANCE according to the installation of Oracle Reports Server. Then run the rwconverter utility and provide the schema password.

```
[oracle@ias6 ~]$ export ORACLE_HOME=/oracle/middleware
[oracle@ias6 ~]$ export ORACLE_INSTANCE=/oracle/middleware/user_projects/domains
/base_domain


[oracle@ias6 ~]$ /oracle/middleware/user_projects/domains/base_domain/reports/bin
/rwconverter.sh userid=CS_190707155545@dbsm-db13 stype=rdffile dtype=rdffile sour
ce=/pub/standard/03.49.30/client/shared/reports/Account\ Interests\ Accruing.rdf
dest=/tmp/Account\ Interests\ Accruing.rdf overwrite=yes compile_all=yes batch=ye
s
Report Builder: Release 12.2.1.2.0 -  Production on Sun Jul 7 16:12:41 2019

Copyright (c) 1982, 2016, Oracle and/or its affiliates.  All rights reserved.

Enter Password:
Converting /pub/standard/03.49.30/client/shared/reports/Account Interests
Accruing.rdf to /tmp/Account Interests Accruing.rdf
```

After few seconds, the new report template will be available in the dest path that is specified in the command line.

Please note:

1. *userid* should be specified in the format oracle_user@DB_TNS_NAME and the Oracle client from ORACLE_HOME should be able to resolve it correctly:

```
[oracle@ias6 ~]$ $ORACLE_HOME/bin/tnsping dbsm-db13

TNS Ping Utility for Linux: Version 11.2.0.3.0 - Production on 07-JUL-2019 16:
19:41
```

```
Copyright (c) 1997, 2011, Oracle.  All rights reserved.

Used parameter files:
/oracle/middleware/network/admin/sqlnet.ora

Used LDAP adapter to resolve the alias
Attempting to contact (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dbsm-db13.spb.
openwaygroup.com)(PORT=1521))(CONNECT_DATA=(SID=DB12C)))
OK (30 msec)
```

2. *batch=yes* parameter is required for running the rwconverter utility from the command line.
3. *compile_all=yes* parameter is required for recompiling RDF to RDF format.
4. All spaces in a report name should be correctly escaped by the backslash symbol.
5. The source report template can be overwritten with the new converted version by specifying *overwrite=yes* parameter.

All reports from the directory can be recompiled and overwritten. The following example shows how:

```
cd $REPORTS_DIR # go to directory which contains report templates
for f in *.[rR][dD][fF] ; do /oracle/middleware/user_projects/domains/base_domain
/reports/bin/rwconverter.sh userid=CS_190707155545/********@dbsm-db13
stype=rdffile dtype=rdffile source="$f" dest="$f" overwrite=yes compile_all=yes
batch=yes; done
```

Note: to avoid asking for the DB connection password for each report, it can be specified in the *userid* parameter. Do not forget to put the source and dest values in double quotes - this will allow reports with spaces in the filename to be converted properly.

Chapter 3. Report Generation Principles

Oracle Reports Server is used to generate reports.

Oracle Reports Builder is used to edit report templates. Reports are generated using the Oracle Reports utility. For more information about working with Oracle reports, refer to documentation for the version of the Oracle Reports Server used.

Oracle Report Definition files (*.rdf) are used to store report templates in WAY4.

Standard report templates are stored in the "<OWS_HOME>/Client/Shared/Reports" directory.

> Standard report templates from the <OWS_HOME> directory can only be modified by the WAY4 vendor and are only upgraded when the WAY4 version is upgraded.

Standard templates based on which reports will be generated require compilation. After compilation, a template is ready to be used and is referred to as a "working" template. Working templates are set up according to the instructions in the section "Installing Oracle Reports Services".

> Working templates must be compiled using the same operating system as that of the workstation on which these reports will be generated.

Working templates are stored in the directory whose path and name is set in the REPORTS_DIRECTORY parameter (see the description of the "REPORTS_DIRECTORY" parameter). Custom working templates created by users are also located in this directory. The directory with working templates should be located either in the report server's file system or on an external file server.

For working templates and custom working templates created independently by users to be stored in separate directories, the "HOME/WORK" option can be used. This option is available if the report server and client application use the same paths to working templates. In this case, working templates compiled on the basis of WAY4 standard templates can be stored in the directory whose path and name is set in the REPORTS_HOME_DIRECTORY parameter (see the description of the "REPORTS_HOME_DIRECTORY" parameter).

For WAY4Web Client, report files are generated in the report server file system or on an external file server. For DB Manager and WAY4 Manager, depending on the settings of the menu item responsible for generating a report, reports may be displayed on the screen, sent to a printer, or saved in a file on a hard disk (in the report server file system or on an external file server).

Processes in WAY4Web Client are monitored using WAY4 Scheduler (see the section document "Scheduler R2"). In DB Manager and WAY4 Manager, report generation for standard templates is registered in the process log. Moreover, errors that occurred during report generation are saved in error logs in the temporary file directory.

Parameters that define rules for generating reports are set in the "dbm.ini" and "db.ini" files (see the section "Parameters Affecting Report Generation").

> Starting from version 03.41.30, the ability to locally generate reports in WAY4 is not supported.

Parameters Affecting Report Generation

This section lists parameters that affect report generation.

The values of the parameters "USE_DBMJ_TO_CALL_REPORTS", "REPORTS_DIRECTORY", "REPORTS_HOME_DIRECTORY", "REPORTS_ENGINE_URL", "REPORT_SERVER_HTTP_TIMEOUT", "CHECK_REPORT_FILE_AVAILABILITY" and "REPORT_SERVER_SSO_CONNECTION", "REPORTS_KRB5_CONF_FILE" are set in the "db.ini" file ([Client.DBM.Params] section).

If reports are generated from DB Manager or WAY4 Manager applications, these parameters can be redefined using the "Database => Configure" system menu item. Note that parameter settings specified in a local machine's client applications will have a higher priority than settings in the "db.ini" configuration file.

USE_DBMJ_TO_CALL_REPORTS

The USE_DBMJ_TO_CALL_REPORTS parameter sets the way reports will be generated:

- "YES" – Reports Engine R2 is used to generate reports.
- "No" – Reports Engine R2 is not used to generate reports.

The default value is "YES".

The parameter works when single reports are generated; reports are generated using WAY4 DB Manager or WAY4 Manager client applications.

REPORTS_DIRECTORY

The "REPORTS_DIRECTORY" parameter sets the path to the (working) directory where working templates created independently by a user or based on standard templates are located.

This parameter is mandatory and must be given a value.

REPORTS_HOME_DIRECTORY

The "REPORTS_HOME_DIRECTORY" parameter sets the path to the (home) directory with working templates based on standard WAY4 templates. If the parameter is set, the Oracle Reports utility searches for a report template in the working directory first ("REPORTS_DIRECTORY"). If the template is not found in the working directory, it is searched for in the home directory.

> This method of storing working templates separately is available if the client application from which a report is run and the Oracle Reports module use the same directory paths. This method cannot be used for storing reports if the client application runs under Windows and the report server runs under Unix.

REPORTS_ENGINE_URL

The "REPORTS_ENGINE_URL" parameter specifies the report server's URL address. Used for HTTP requests when generating reports.

This parameter is mandatory and must be given a value.

REPORT_SERVER_HTTP_TIMEOUT

The "REPORT_SERVER_HTTP_TIMEOUT" parameter sets the maximum time to wait for a response from the report server.

The parameter value is specified in seconds, the default value is 3600 seconds.

CHECK_REPORT_FILE_AVAILABILITY

The "CHECK_REPORT_FILE_AVAILABILITY" parameter indicates whether a check should be made for a working template. This parameter is used if the "HOME/WORK" option is used (see the description of the REPORTS_HOME_DIRECTORY parameter).

By default the parameter is not used and its value is "No".

REPORT_SERVER_SSO_CONNECTION

The "REPORT_SERVER_SSO_CONNECTION" parameter sets the user identification method if Single Sign-On mode is used (SSO, single identification in a virtual infrastructure).

This parameter is mandatory in SSO mode.

REPORTS_KRB5_CONF_FILE

The "REPORTS_KRB5_CONF_FILE" parameter sets the full name of the configuration file for the Kerberos network authentication protocol.

This parameter is mandatory in SSO mode.

Report Batch Execution

The pipe "com.openwaygroup.pipe.executing_report_batch.jar" is used to generate batch reports, see "Appendix 1. "Executing Report Batch" Java Pipe Parameters.

Local Language Support in Report Generation

WAY4 supports the ability provided by Oracle Reports to generate reports in different languages.

To use this feature, do as follows:

- The language must be registered in the "Languages" system dictionary (for WAY4Web Client this is the menu item "System core → Localization → Languages"; for DB Manager and WAY4 Manager – "Full → Configuration Setup → Language and Localisation → Languages").

The "Languages" dictionary contains the list of languages that can be used in WAY4 (for more information, see the section "Languages" dictionary" of the document "Dictionaries ").

- An XML file with a list of report element names in the local language must be created (custom XML file, Customize XML). This file may have the same name as the main report template but with the "*.xml" extension.

The encoding in which report elements will be shown is specified in the XML file header using the "encoding" parameter, for example, encoding="UTF-8".

- Convert the custom XML file into an Oracle Report Definition file (*.rdf). The "rwconverter" utility included in the Oracle Reports distribution must be used for conversion. For details on using the utility, see documentation for the version of the Oracle Reports generator used. The converted file must be stored in the directory specified in the REPORTS_DIRECTORY parameter.
- In the *Report Name* field of the parameters of the "Oracle Report" menu sub-item calling the report, specify the name of the report file (*.rdf) translated into the local language (see the section

  "Oracle Report" Type" of the document "Menu Editor").
- If necessary, translate client messages that will be included in the report into the local language (see the section "Translating Messages into Local Languages" of the document "Configuration of Client Messages").

For DB Manager and WAY4 Manager applications the parameter defining the language of report content can be set in the following ways (listed in descending priority; if the report language was set using two or more parameters, the value of the parameter with the higher priority will be used):

- With the "P_LANGUAGE" parameter of the "Oracle Report" menu item calling the report. This setting has the highest priority.
- In generating a report, the system user's language can be used. This language is set in the *Language* field of the "Constants for <user group name>" form opened by clicking the [Constants] button in the "User Groups and Users - View" grid form (Full → DB Administrator Utilities → Users & Grants → User Groups and Users – View).
- In generating a report, the language specified in WAY4 global settings can be used ("Full → Configuration Setup → Main Tables → Global Constants").

The current solution does not support the ability to redefine the report generation language for WAY4Web Client.

For reports provided to clients (statements) the client's language can be used to determine the report language.

Sample setup:

- German is registered in the WAY4 database (see Fig. 1).

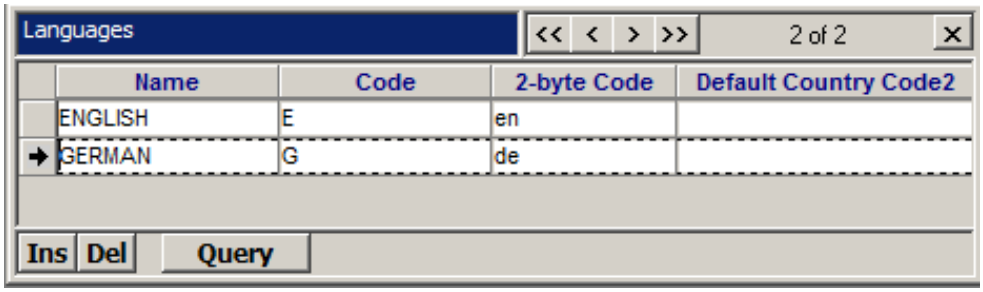

Fig. 1. List of languages registered in WAY4

- In the "Test_Report_German.xml" XML file, labels are translated into German. A fragment of this file containing a translation of the "B_Field" object (contract number) into German ("Feld") is shown below.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<report name="Test_Report" DTDVersion="9.0.2.0.10">

<customize>

...

<object name="B_Field" type="REP_GRAPHIC_TEXT">

<properties>

<property name="textSegment"><![CDATA[Feld]]></property>

</properties>

</object>

...

</customize>

</report>
```

- Using the "rwconverter" utility, the "Test_Report_German.rdf" file was created and put into the working directory; the path to this directory is specified in the "REPORTS_DIRECTORY" parameter.
- Messages that will be included in the report are translated into German.
- For the menu item that calls this report, "Test_Report_German.rdf" is specified in the *Report Name* field of "Oracle Report" menu sub-item parameters.
- The *Languages* field in "Global Constants" is empty.
- "GERMAN" is specified in the *Language* field of the "Constants for German-Speaking Staff" form. For other user groups, the *Language* field is not filled in.
- The "P_LANGUAGE" parameter is not defined for the menu item that calls this report.

With these settings, users in the "German-Speaking Staff" group will get reports with labels in German and that contain client messages translated into German.

If a German is specified in the "P_LANGUAGE" parameter of the menu item that calls this report, the report will be generated in German for all users.

Chapter 4. "Executing Report Batch" Java Pipe Parameters

The pipe is used with the following parameters:

Parameter	Value	Mandatory /Optional	Description
SQL_SELECT	SQL SELECT script	Mandatory	SQL query text. For each string returned, a query is sent to the Oracle Reports utlity. Columns specified in the query can be used as pipe parameters. Example of columns used in queries: STATUS_TEXT = FinInst: %8%, Branch: %1%, Report Type: %2% P_ZIP = @ZIP_CODE@ Column names or the serial number of a column in the table can be specified as query parameter values. The name or number of a column must be framed by the '@' character (the '%' character is used for compatibility).
STATUS_TEXT		Mandatory	Text that will be shown in the status panel when each job is generated.
INI_SECTION		Optional	Name of the section in the db.ini file with additional parameters that will be transferred to the Oracle Reports utility. Used to generate more than one report with a set of specific parameters.
reportName, REPORT		Mandatory	Name of the report file (RDF). The path to the report file is not specified in the parameter.
destName, DESNAME		Mandatory	Name of the result file (with reports) or email address for sending the result file.
			Report destination: "file" – the file is saved on the client computer. "server_file" – the file is saved on the report server.

Parameter	Value	Mandatory /Optional	Description
destType	file/server_file /mail/printer	Optional	"mail" – the report file is sent from the report server to an email address. "printer" – the report file is sent from the report server to a printer. The default value is "file".
processTitle, TITLE		Optional	Report header transferred to the Oracle Reports utility in the P_TITLE parameter.
destFormat, DESFORMAT	DFLT/HTML/PDF /SPREADSHEET /RTF/XML /wide/...	Optional	Oracle Reports utility parameter defining the file format. The default value is "wide". See Oracle documentation for a detailed description of the parameter (http://www.oracle.com).
drawMode, MODE	bitmap /character	Optional	Oracle Reports utility parameter defining the file type. The default value is "character". See Oracle documentation for a detailed description of the parameter (http://www.oracle.com).
numberOfCopies		Optional	Oracle Reports utility parameter defining the number of file copies. The default value is "1". See Oracle documentation for a detailed description of the parameter (http://www.oracle.com).
REPORTS_ENGINE_URL		Optional	The parameter redefines the value of the REPORTS_ENGINE_URL parameter (report server URL address).
P_*		Optional	Parameters whose names begin with "P_" are transferred to the report server as is.