

## **Oracle® Insurance Calculation Engine (OICE)**

### **Databases**

#### **Installation Instructions – Step 1**

Version 9.7.0.0

Documentation Part Number: E51103-01

December, 2013

Copyright © 2009, 2013, Oracle and/or its affiliates. All rights reserved.

#### Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

#### **License Restrictions**

##### Warranty/Consequential Damages Disclaimer

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.

##### Warranty Disclaimer

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.

##### Restricted Rights Notice

If this is software or related documentation that 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 END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

##### Hazardous Applications Notice

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

##### Third Party Content, Products, and Services Disclaimer

This software or hardware 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.

## Table of Contents

<b>OVERVIEW</b> .....	4
<b>AUDIENCE</b> .....	4
<b>CUSTOMER SUPPORT</b> .....	4
<b>PREREQUISITES</b> .....	4
<b>STEPS TO INSTALL THE DATABASE</b> .....	5
Oracle 11g.....	5
Run SQL*Plus.....	5
Create Users.....	5
Import the Databases .....	5
Create a Read-Only Database User.....	6
Microsoft SQL Server 2005.....	7
Create New Databases .....	7
Create Database Users .....	7
Restore the Databases .....	7
IBM DB2 .....	8
Configure DB2 .....	8
Create the Database .....	8
Create Users.....	8
Configure the Databases.....	8
Prepare the DDL Script .....	9
Create the Schema Using the DDL Script.....	9
Load the Database Data.....	9
Import Client Data.....	9
<b>APPENDIX</b> .....	10

## OVERVIEW

The Oracle Insurance Calculation Engine (OICE) application and the Rules Palette application together form a complete solution. A four step installation process is required in order to install and set-up both applications. These instructions represent step one of that process. Refer to the documentation library included with this release for the other three steps of the installation.

This installation guide is meant solely for the installation of a development environment and installations for production environments many require additional/different configuration.

## AUDIENCE

These instructions are written for Database Administrators who have comprehensive knowledge of creating and configuring databases.

## CUSTOMER SUPPORT

If you have any questions about the installation or use of our products, please visit the My Oracle Support website: <https://support.oracle.com>, or call (800) 223-1711.

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>

or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## PREREQUISITES

- One of the following databases:
  - Oracle 11g (11.2.0. 3)
  - Microsoft SQL Server 2008 R2 SP2
  - DB2 9.7 Fix Pack 7
- Any compatible operating system
- Oracle Insurance Calculation Engine V9.7.0 Media Pack from the Oracle Software Delivery Cloud.  
Select the Media pack that corresponds to the application server you will be using.
- Database from the Oracle Insurance Calculation Engine V9.7.0 Media Pack on the Oracle Software Delivery Cloud. The database information is listed as a separate download within the Media Pack.

## STEPS TO INSTALL THE DATABASE

Install the database according to the vendor's instructions. Be sure to note the database installation user name and password for use later in the installation process.

### Oracle 11g

The Oracle database must use a Unicode character set defined at database creation. Complete the following installation tasks:

#### Run SQL\*Plus

Run SQL\*Plus as a user with DBA privileges by running the following command:

```
sqlplus / as sysdba
```

#### Create Users

1. Create the Oracle users for the OICE and IVS schemas by entering the following commands from a sqlplus session:

```
create user <OICE Username> identified by <OICE Password>;
grant connect, resource to <OICE Username>;
grant create view to <OICE Username>;
grant create trigger to <OICE Username>;

create user <IVS Username> identified by <IVS Password>;
grant connect, resource to <IVS Username>;
grant create view to <IVS Username>;
grant create trigger to <IVS Username>;
```

#### Import the Databases

1. Create the Oracle directories for the fully qualified path to the export data pump dumpfile. Enter the following commands from a sqlplus session:

```
create directory oice_dir as '<fully qualified path holding the dumpfiles>'; (Example
/opt/oracle/backups/)
grant read, write on directory oice_dir to <OICE Username>;
grant read, write on directory oice_dir to <IVS Username>;
exit;
```

2. Create and load the OICE and IVS databases using import data pump (impdp):

---

**Note:** The impdp command can be run interactively. It will prompt you to provide any required arguments not on the command line, or you can enter them in one command.

The system password is the same password entered when the database was configured, and the OICE and IVS Usernames and passwords are those entered in the Create Users section above.

---

```
impdp system/<system password>
directory=oice_dir
dumpfile=OICE.dmp
logfile=OICE_PAS.log
full=yes
remap_schema=OICE:<OICE_Username>

impdp system/<system password>
directory=oice_dir
dumpfile=oiceivs.dmp
logfile=OICE_IVS.log
full=yes
remap_schema=oiceivs:<IVS_Username>
```

## Create a Read-Only Database User

A shell script to create a read-only database user is provided in the Appendix of this installation guide. The shell script is used instead of creating the user manually, as there are several complex options and privileges that must be configured. Make sure the script is executed as the Oracle user in a UNIX shell.

---

**Note:** A Windows version of this script is not available.

---

1. Copy the contents of the appendix and paste it into a text file.
2. Save the file as `Create_READONLY_user.sh`. Make sure the file is saved with UNIX line terminations.
3. Change the permissions so the script can be executed by the shell by entering the following command:  
`chmod 770 Create_READONLY_user.sh`
4. Execute the script by entering the following command:  
`./Create_READONLY_user.sh`
5. When prompted you will need to answer the following:
  - SYSTEM password – The password that was set during the configuration of Oracle 11g.
  - Table Owner Userid – Type the OICE Username used above.
  - Read Only Userid – Type **OICE\_RO**. You may use this example or create your own user.
  - Read Only Password – Type **OICE\_RO**. You may use this example or create your own password.

The script will run and display the results of creating the read-only user.

## Microsoft SQL Server 2005

The Microsoft SQL Server Management Studio is used to create the new databases and users, as well as restore the databases from the provided backups.

### Create New Databases

Using Microsoft SQL Server Management Studio, create two new databases: one for the OICE data, and one for the IVS data.

### Create Database Users

Three new database users will need to be created: two for the OICE database and one for the IVS database.

- OICE database user with full privileges for the OICE database
- OICE database user with read-only privileges for the OICE database
- IVS database user with full privileges for the IVS database

### Restore the Databases

The OICE Media Pack includes backups of both databases that can be restored using Microsoft SQL Server Management Studio. To restore the databases, right-click on the specific database in SQL Server Management Studio and select **restore**. Ensure that the correct backup file is used to restore each database.

- OICE database: oice\_pas.bak
- IVS database: oice\_ivs.bak

## IBM DB2

### Configure DB2

Several database settings must be configured prior to creating the OICE databases. To configure the database settings, use the db2 and db2set commands for DB2:

- Global Settings

```
db2set DB2_USE_ALTERNATE_PAGE_CLEANING=YES
db2set DB2_REDUCED_OPTIMIZATION=TRUE
db2set DB2_EVALUNCOMMITTED=TRUE
db2set DB2_CORRELATED_PREDICATES=YES
db2set DB2_SKIPINSERTED=YES
db2set DB2_SKIPDELETED=YES
```

- DBM Changes

```
db2 update dbm cfg using sheapthres 120000
db2 update dbm cfg using mon_heap_sz 256
db2 update dbm cfg using query_heap_sz 2048
```

### Create the Database

A database with two schemas will need to be created; one schema for OICE data and one for IVS data.

Use the **db2 create** command to create the database.

```
db2 create database OICE
```

### Create Users

Three database users must be created:

- OICE\_PAS — OICE user with full privileges
- OICE\_RO — OICE user with read-only privileges
- OICE\_IVS — IVS user with full privileges

Since DB2 uses the operating system for authentication, these users must first be created at the operating system level. Please consult the operating system documentation for creating users.

### Configure the Databases

The database settings for the database must be configured.

```
db2 connect to OICE
db2 update db cfg using dbheap 2400
db2 update db cfg using logbufsz 512
db2 update db cfg using locklist 10000
db2 update db cfg using app_ctl_heap_sz 256
db2 update db cfg using sortheap 1024
db2 update db cfg using applheapsz 4096
db2 update db cfg using locktimeout 360
db2 update db cfg using maxlocks 76
db2 update db cfg using chngpgs_thresh 30
db2 update db cfg using num_iocleaners 7
```

```
db2 update db cfg using num_ioservers 7
db2 update db cfg using logfilsiz 20000
db2 update db cfg using logprimary 30
db2 update db cfg using logsecond 0
db2 update db cfg using pckcachesz 1024
db2 update db cfg using catalogcache_sz 512
db2 update db cfg using maxfilop 256
db2 update db cfg using maxappls 60
db2 update db cfg using avg_appls 1
db2 update db cfg using PCKCACHESZ 2048
db2 update db cfg using SORTHEAP 512
db2 update db cfg using dft_queryopt 3
```

## Prepare the DDL Script

The `db2look_oipa_pas.ddl` and the `OICE_IVS.ddl` files must be edited to include the fully-qualified path names for each tablespace creation command.

## Create the Schema Using the DDL Script

The `db2look_oipa_pas.ddl` file and the `OICE_IVS.ddl` file, which were modified in the previous step, will now be used to create the database schemas.

```
db2 -tvf db2look_oipa_pas.ddl >db2look_oice_pas.log
db2 -tvf OICE_IVS.ddl >db2look_oice_ivs.log
```

After execution has completed, reference the created log files, `db2look_oice_pas.log` and `db2look_oice_ivs.log`, to ensure the schemas were successfully created.

## Load the Database Data

The `db2move` command will be used to load the data into the database.

To load the OICE database, first ensure that you are currently in the directory that contains the OICE database data from the Oracle Insurance Calculation Engine Media Pack.

```
db2move OICE load
```

To load the IVS database, first ensure that you are currently in the directory that contains the IVS database data from the Oracle Insurance Calculation Engine Media Pack.

```
db2move OICE load
```

## Import Client Data

After the database load is complete, run the associated `AsClientInsert.sql` file to insert the required `AsClient` records.

## APPENDIX

```

#!/bin/bash
#-----
#-- This script creates a new user and grants    --
#-- read privileges for the objects of the      --
#-- chosen owner.                                --
#-----
#-- The script is run from the linux command    --
#-- line. The linux user must have sqlplus      --
#-- access to the database -- so environment    --
#-- variables must be set                      --
#--                                           --
#-- The script is interactive and will request: --
#--     password for the "SYSTEM" user          --
#--     Userid of the owner of the objects      --
#--     Userid of read only user to be created  --
#--     Password of the read only user to be    --
#--         created                           --
#-----
#-- The script runs an inline process that:      --
#--     Creates the read only user              --
#--     Creates a role                         --
#--     Grants connect and the new role to the  --
#--         read only user                   --
#--     Loops through the objects of the owner  --
#--     Grants select on tables and views to the --
#--         new role                         --
#--     Grants execute on the procedures,      --
#--         functions and packages to the new role--
#--     Creates a synonym for each object granted --
#-----

#-- Interact with the user to get the system password
#-- object owner user , read user and read password

echo ''
read -s -p "Enter SYSTEM Password: " spass
echo ''
read -p "Enter Table Owner Userid: "ouser
echo ''
read -p "Enter Read Only Userid: "ruser
read -s -p "Enter Read Only Password: "rpass
echo ''
echo ''

#-- Formulate the role name from the owning user

export rorole=${ouser}_role

#-- Start a sqlplus session as system

sqlplus -S system/${spass} <<EOF

-- Set sqlplus variables

set echo off
set heading off
set verify off
set serveroutput on

```

```

-- Set session variables to hold entered information

varouser varchar2(30);
defineouser = ${ouser};
varruser varchar2(30);
defineruser = ${ruser};
varrpass varchar2(30);
definerpass = ${rpass};
varrorole varchar2(60);
definerorole = ${rorole};

-- Start inline procedure

declare

-- Procedure variables

v0User varchar2(30) := upper('&ouser');
vRUser varchar2(30) := upper('&ruser');
vRoRole varchar2(60) := upper('&rorole');
Cnt number;
SQLstmt varchar2(500);
begin

-- Check if read only user exists

    select count(*) into Cnt
        from dba_users
        where username = vRUser;

-- if read only user does not exist create new

    if Cnt < 1 then
        SQLstmt := 'CREATE USER ' || '&ruser' || ' IDENTIFIED BY ' || '&rpass';
        execute immediate SQLstmt;
    else
        dbms_output.put_line(vRUser || ' ALREADY EXISTS, GRANTS WILL BE RE-EXECUTED');
    end if;

-- See if role exists

    select count(*) into Cnt
        from dba_roles
        where role = vRoRole;

-- If role does not exist create role

    if Cnt < 1 then
        SQLstmt := 'CREATE ROLE ' || vRoRole;
        execute immediate SQLstmt;
    end if;

-- Grant connect and new role to read only user

    SQLstmt := 'GRANT CONNECT, ' || '&rorole' || ' TO ' || '&ruser';
    execute immediate SQLstmt;

-- loop through the object owners objects and grant access to
-- the read only user and create synonyms

    for inrec in (select object_type, object_name
                  from dba_objects
                  where owner = v0User) loop

```

```

-- if the object_type is table or view grant select to read user

    if inrec.object_type = 'TABLE' or inrec.object_type = 'VIEW' THEN
        SQLStmt := 'GRANT SELECT ON ' || '&ouser' || '.' || inrec.object_name || ' TO ' ||
'&rorole';
        execute immediate SQLStmt;

-- Check if synonym exists

    select count(*) into Cnt
    from dba_synonyms
    where owner = vRUser
    and synonym_name = inrec.object_name;

-- Create synonym if one does not exist

    if Cnt < 1 then
        SQLStmt := 'CREATE SYNONYM ' || '&ruser' || '.' || inrec.object_name || ' FOR ' ||
'&ouser' || '.' || inrec.object_name;
        execute immediate SQLStmt;
    end if;
    else

-- Check if object type is procedure, function or package and grant execute

    if inrec.object_type = 'PROCEDURE'
    or inrec.object_type = 'FUNCTION'
    or inrec.object_type = 'PACKAGE' THEN
        SQLStmt := 'GRANT EXECUTE ON ' || '&ouser' || '.' || inrec.object_name || ' TO ' ||
'&rorole';
        execute immediate SQLStmt;

-- Check if synonym exists

    select count(*) into Cnt
    from dba_synonyms
    where owner = vRUser
    and synonym_name = inrec.object_name;

-- Create synonym if one does not exist

        if Cnt < 1 then
            SQLStmt := 'CREATE SYNONYM ' || '&ruser' || '.' || inrec.object_name || ' FOR ' ||
'&ouser' || '.' || inrec.object_name;
            execute immediate SQLStmt;
        end if;
        end if;
        end if;
    end loop;
end;
/
exit;
EOF

```