

# **Oracle® Insurance Policy Administration**

## **Databases**

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

Version 9.4.0.0

Documentation Part Number: E18894\_01

June 2011

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

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

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

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

## **U.S. GOVERNMENT RIGHTS**

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

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

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

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

# TABLE OF CONTENTS

<b>OVERVIEW .....</b>	<b>4</b>
AUDIENCE .....	4
CUSTOMER SUPPORT .....	4
PREREQUISITES .....	4
<b>STEPS TO INSTALL THE DATABASE .....</b>	<b>5</b>
ORACLE 11G .....	5
Run SQL*Plus .....	5
Create Users .....	5
Import the Databases .....	6
Create a Read-Only Database User .....	6
MICROSOFT SQL SERVER 2005 .....	8
Create New Databases .....	8
Create Database Users .....	8
Restore the Databases .....	8
IBM DB2 .....	9
Configuring DB2 .....	9
Create the Database .....	9
Create Users .....	9
Configure the Databases .....	10
Prepare the DDL Script .....	11
Create the Schema Using the DDL Script .....	11
Load the Database Data .....	11
Grant User Privileges .....	12
<b>APPENDIX .....</b>	<b>14</b>

## OVERVIEW

The Oracle Insurance Policy Administration (OIPA) 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.

## PREREQUISITES

- Oracle 11gR2(11.2), Microsoft SQLServer 2005(9.00.5000.00) or DB2 9.7 Fix Pack 3a database
- Any compatible operating system
- Oracle Insurance Policy Administration V9.4.0 Media Pack from E-Delivery. Select the Media pack that corresponds to the application server you will be using.
- Database from the Oracle Insurance Policy Administration V9.4.0 Media Pack on E-Delivery. 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.

### ORACLE 11G

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

#### Run SQL\*Plus

1. 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 OIPA and IVS schemas by entering the following commands from a sqlplus session:

```
create user <OIPA Username> identified by <OIPA Password>;  
grant connect, resource to <OIPA Username>;  
grant create view to <OIPA Username>;  
grant create trigger to <OIPA 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 oipa_dir as '<fully qualified path holding the
dumpfiles>'; (Example /opt/oracle/backups/)
grant read, write on directory oipa_dir to <OIPA Username>;
grant read, write on directory oipa_dir to <IVS Username>;
exit;
```

2. Create and load the oipa and IVS databases using import data pump (impdp);

```
impdp system/<system password>
  directory=oipa_dir
  dumpfile=OIPA_PAS.dmp
  logfile=OIPA_PAS.log
  exclude=grant
  full=yes
  remap_schema=oipa:<OIPA Username>
```

```
impdp system/<system password>
  directory=oipa_dir
  dumpfile=OIPA_IVS.dmp
  logfile=OIPA_IVS.log
  exclude=grant
  full=yes
  remap_schema=gaivs:<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.

1. Change the authority so the script can be executed by Oracle by entering the following command:

```
chmod 770 Create_readonly_user.sh
```

2. Execute the script by entering the following command:

```
./Create_readonly_user.sh
```

3. 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 **OIPA Username**.
  - Read Only Userid – Type **OIPA\_RO**. You may use this example or create your own user.
  - Read Only Password – Type **OIPA\_RO**. You may use this example or create your own password.

## 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 OIPA data, and one for the IVS data.

### Create Database Users

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

1. OIPA database user with full privileges for the OIPA database
2. OIPA database user with read-only privileges for the OIPA database
3. IVS database user with full privileges for the IVS database

### Restore the Databases

The OIPA 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.

1. OIPA database: oipa\_pas.bak
2. IVS database: oipa\_ivs.bak



## IBM DB2

### Configuring DB2

Several database settings must be configured prior to creating the OIPA databases. To configure the database settings, the DB2 db2 and db2set commands are used.

#### 1. 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`

#### 2. 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 OIPA data and one for IVS data.

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

- `db2 create database <Database Name>`

### Create Users

Three database users must be created:

- OIPA\_PAS -- OIPA user with full privileges
- OIPA\_RO -- OIPA user with read-only privileges
- OIPA\_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.

1. db2 connect to <Database Name>
2. db2 update db cfg using dbheap 2400
3. db2 update db cfg using logbufsz 512
4. db2 update db cfg using locklist 10000
5. db2 update db cfg using app\_ctl\_heap\_sz 256
6. db2 update db cfg using sortheap 1024
7. db2 update db cfg using applheapsz 4096
8. db2 update db cfg using locktimeout 360
9. db2 update db cfg using maxlocks 76
10. db2 update db cfg using chngpgs\_thresh 30
11. db2 update db cfg using num\_iocleaners 7
12. db2 update db cfg using num\_ioservers 7
13. db2 update db cfg using logfilsiz 20000
14. db2 update db cfg using logprimary 30
15. db2 update db cfg using logsecond 0
16. db2 update db cfg using pckcachesz 1024
17. db2 update db cfg using catalogcache\_sz 512
18. db2 update db cfg using maxfilop 256
19. db2 update db cfg using maxappls 60
20. db2 update db cfg using avg\_appls 1
21. db2 update db cfg using PCKCACHESZ 2048
22. db2 update db cfg using SORTHEAP 512
23. db2 update db cfg using dft\_queryopt 3

## Prepare the DDL Script

The `db2look_oipa_pas.ddl` and the `db2look_oipa_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 `db2look_oipa_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_oipa_pas.log
db2 -tvf db2look_oipa_ivs.ddl >db2look_oipa_ivs.log
```

After execution has completed, reference the created log files, `db2look_oipa_pas.log` and `db2look_oipa_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 OIPA database, first ensure that you are currently in the directory that contains the OIPA database data from the Oracle Insurance Policy Administration Media Pack.

```
db2move <Database Name> 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 Policy Administration Media Pack.

```
db2move <Database Name> load
```

## Grant User Privileges

### OIPA User

1. To create the statement to grant the OIPA\_PAS user ALL and EXECUTE privileges, execute the following SQL statement:

```
GO' FROM SYSIBM.SYSTABLES WHERE CREATOR = 'OIPA_PAS'
UNION
SELECT 'GRANT EXECUTE ON PROCEDURE OIPA_PAS.' || procname || ' TO
USER OIPA_PAS
GO' FROM SYSIBM.SYSPROCEDURES WHERE PROCSHEMA = 'OIPA_PAS'
UNION
SELECT 'GRANT EXECUTE ON FUNCTION OIPA_PAS.' || name || ' TO USER
OIPA_PAS
GO' FROM SYSIBM.SYSFUNCTIONS WHERE SCHEMA = 'OIPA_PAS'
```

2. Execute the results of the previous SELECT statement in step 1 to grant the OIPA user ALL AND EXECUTE privileges for all tables, stored procedures, and functions.

### Read-only OIPA User

1. To create the SQL statement to grant the read-only OIPA\_PAS user SELECT privileges, execute the following SQL statement:

```
SELECT 'GRANT SELECT ON OIPA_PAS.' || name || ' TO USER OIPA_RO
GO' FROM SYSIBM.SYSTABLES WHERE CREATOR = 'OIPA_PAS'
```

2. Execute the results of the previous SELECT statement in step 3 to grant the read-only user SELECT privileges.

**Note:** If a new table is added to the database, then the Read Only OIPA\_PAS user scripts above must be run again.

## IVS User

1. To create the SQL statement to grant the OIPA\_IVS user ALL privileges, execute the following SQL statement:

```
SELECT 'GRANT SELECT ON OIPA_IVS.' || name || ' TO USER OIPA_IVS  
GO' FROM SYSIBM.SYSTABLES WHERE CREATOR = 'OIPA_IVS'
```

2. Execute the results of the previous SELECT statement in step 5 to grant the IVS user ALL privileges.

## APPENDIX

```
#!/bin/bash

#-----
#-- This script creates a new user and grants      --
#-- read privileges for the objects of the        --
#-- choosen owner.                                --
#-----
#-- The script is run from the linux command      --
#-- line. The linux user must have sqlplus        --
#-- access to the databaase -- 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 theprocedures,              --
#-- 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} <<EOFA

-- Set sqlplus valraibles

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

-- Set session variables to hold entered information

var ouser varchar2(30);
define ouser = ${ouser};
var ruser varchar2(30);
define ruser = ${ruser};
var rpass varchar2(30);
define rpass = ${rpass};
var rorole varchar2(60);
define rorole = ${rorole};

-- Start inline procedure

declare

--    Procedure variables

    vOUser 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, ' || '&rrole' || ' 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 = vOUser) 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
```