

# **Oracle® Insurance Policy Administration**

## **Databases**

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

Version 9.3.1.0

Documentation Part Number: E21044-01

March 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>  |
| CUSTOMER SUPPORT .....                                | 4         |
| PREREQUISITES.....                                    | 4         |
| <b>STEPS TO INSTALL THE DATABASE.....</b>             | <b>5</b>  |
| ORACLE 11G.....                                       | 5         |
| Run SQL*Plus .....                                    | 5         |
| Create Tablespaces .....                              | 5         |
| Create the Users and Grant Privileges.....            | 6         |
| Create a Read-Only Database User.....                 | 6         |
| Import the Databases.....                             | 7         |
| Run the Indexing Script .....                         | 7         |
| MICROSOFT SQL SERVER 2005 .....                       | 8         |
| Create a New Database .....                           | 8         |
| Restore the Databases .....                           | 8         |
| IBM DB2 .....   | 9         |
| Configuring DB2 .....                                 | 9         |
| Create the Databases .....                            | 9         |
| Configure the Databases.....                          | 11        |
| Prepare the DDL Script.....                           | 12        |
| Create the Schema Using the DDL Script .....          | 12        |
| <b>APPENDIX A: ORACLE READ-ONLY USER SCRIPT .....</b> | <b>13</b> |
| <b>APPENDIX B: ORACLE INDEXING SCRIPT .....</b>       | <b>16</b> |
| <b>APPENDIX C: WEB SERVICE SECURITY SCRIPTS .....</b> | <b>21</b> |
| SQL SERVER DATABASE SCRIPTS .....                     | 22        |
| ORACLE DATABASE SCRIPTS.....                          | 23        |
| DB2 DATABASE SCRIPTS.....                             | 24        |

## 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.

## 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 11g, Microsoft SQLServer 2005 or DB2 database
- Any compatible operating system
- Oracle Insurance Policy Administration V9.3.1 Media Pack from E-Delivery.
- Database from the Oracle Insurance Policy Administration V9.3.1 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 Tablespaces

1. Create the OIPA\_PAS tablespace for OIPA by entering the following command in SQL\*Plus:

```
CREATE TABLESPACE OIPA_PAS  
DATAFILE '<FULL PATH OF THE DATA FILE>'  
SIZE 100M  
AUTOEXTEND ON  
MAXSIZE 2000M  
EXTENT MANAGEMENT LOCAL;
```

2. Create the OIPA\_IVS tablespace for IVS by entering the following command in SQL\*Plus:

```
CREATE TABLESPACE OIPA_IVS  
DATAFILE '<FULL PATH OF THE DATA FILE>'  
SIZE 100M  
AUTOEXTEND ON  
MAXSIZE 2000M  
EXTENT MANAGEMENT LOCAL;
```

## Create the Users and Grant Privileges

1. Create the OIPA user and grant privileges. In the sample command below the username and password are set to OIPA. These may be changed if necessary.

```
CREATE USER OIPA
IDENTIFIED BY OIPA
DEFAULT TABLESPACE OIPA_PAS;
GRANT CONNECT, RESOURCE TO OIPA;
```

2. Create the IVS user. In the sample command below the username and password are set to IVS. This may be changed if necessary.

```
CREATE USER IVS
IDENTIFIED BY IVS
DEFAULT TABLESPACE OIPA_IVS;
GRANT CONNECT, RESOURCE TO IVS;
```

3. Exit SQL\*Plus to return to the command prompt by entering the following command:  
EXIT;

## Create a Read-Only Database User

A shell script to create a read-only database user is provided in Appendix A of this document. When creating the script file, name it `Create_readonly_user.sh`. 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. Load the attached script using the following command and keystrokes:

```
vi Create_readonly_user.sh
A
<paste the scripts in>
:wq
```

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

```
chmod 770 Create_readonly_user.sh
```

3. Execute the script by entering the following command:

```
./Create_readonly_user.sh
```

4. 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**, if you followed the user example from above.
  - 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.

## Import the Databases

Enter the following commands at a shell prompt to import the OIPA and IVS databases:

```
imp <OIPA Username>/<password> file=oipa_pas_9_3.dmp full=yes  
imp <IVS Username>/<password> file=oipa_ivs_9_3.dmp full=yes
```

## Run the Indexing Script

In order to enable case-insensitive searching, you must run the SQL script included in Appendix B using SQL\*Plus or any other database querying client. The script should be run as the OIPA database 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 a New Database

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 Databases

Two databases will need to be created, one for the OIPA data and one for IVS data.

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

- `db2 create database asadmin`
- `db2 create database asivs`

### Create Users

Three database users must be created:

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

## Grant User Privileges

### OIPA User

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

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

2. Execute the results of the previous SELECT statement in step 1 to grant the OIPA user ALL privileges.

### Read-only OIPA User

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

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

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

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

### IVS User

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

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

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

## Configure the Databases

The database settings for each of the two databases must be configured. The following steps should be completed for both the OIPA and IVS database.

1. db2 connect to asadmin (the correct database name must be specified)
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_asadmin.out` and `db2look_asivs.out` 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_asadmin.out` file, which was modified in the previous step, will now be used to create the database schema.

```
db2 -tvf db2look_asadmin.out >db2look_asadmin.log
db2 -tvf db2look_asivs.out >db2look_asivs.log
```

After execution has completed, reference the created log files, `db2look_asadmin.log` and `db2look_asivs.log`, to ensure the schema was 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 asadmin 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 asivs load
```

## APPENDIX A: ORACLE READ-ONLY USER SCRIPT

This script will allow you to add a read-only user to the database. This can only be used for Oracle 11g databases running on Linux, Solaris and AIX.

The script below must be executed as the Oracle user in a UNIX shell.

**Note:** This script should be run each time a new stored procedure is created in the Rules Palette.

```
*****
#!/bin/bash

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 ''
export rorole=${ouser}_readonly_role
sqlplus -S system/${spass} <<EOF
set echo off
set heading off
set verify off
set serveroutput on

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};

declare
    vOUser varchar2(30) := upper('&ouser');
    vRUser varchar2(30) := upper('&ruser');
    vRoRole varchar2(60) := upper('&rorole');
```

```

        Cnt number;
        SQLstmt varchar2(500);
begin

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

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

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

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

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

    for inrec in (select table_name
        from dba_tables
        where owner = vOUser
        and table_name like 'AS%'
        and table_name <> 'ASSECURITY') loop

        SQLstmt := 'GRANT SELECT ON ' || '&ouser' || '.' || inrec.table_name
||
' TO ' || '&rrole';
        execute immediate SQLstmt;

        select count(*) into Cnt
            from dba_synonyms
            where owner = vRUser

```

```

        and synonym_name = inrec.table_name;

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

    end loop;
end;
/
exit;
EOF
*****

```

## APPENDIX B: ORACLE INDEXING SCRIPT

This script creates indexes for each table that allow for case-insensitive searching. You must run this script after importing the database.

```
CREATE INDEX CI_1_ASACTIVITY
    ON ASACTIVITY(NLSSORT("ACTIVITYGUID", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("TRANSACTIONGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_1_ASACTIVITYFIELD
    ON ASACTIVITYFIELD(NLSSORT("FIELDNAME", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ACTIVITYGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_1_ASADDRESS
    ON ASADDRESS(NLSSORT("ADDRESSGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_2_ASADDRESS
    ON ASADDRESS(NLSSORT("STATECODE", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ADDRESSGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_1_ASADDRESSFIELD
    ON ASADDRESSFIELD(NLSSORT("FIELDNAME", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ADDRESSGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_ASADDRESSROLE
    ON ASADDRESSROLE(NLSSORT("DEFAULTFLAG", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("CLIENTGUID", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ADDRESSGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_1_ASCLIENT
    ON ASCLIENT(NLSSORT("LASTNAME", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("CLIENTGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO

CREATE INDEX CI_2_ASCLIENT
```



```

        ON ASCLIENT(NLSSORT("FIRSTNAME",'nls_sort=''BINARY_CI'''),
NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_3_ASCLIENT
        ON ASCLIENT(NLSSORT("TAXID",'nls_sort=''BINARY_CI'''),
NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_4_ASCLIENT
        ON ASCLIENT(NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASCLIENTFIELD
        ON ASCLIENTFIELD(NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_2_ASCLIENTFIELD
        ON ASCLIENTFIELD(NLSSORT("FIELDNAME",'nls_sort=''BINARY_CI'''),
NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("TEXTVALUE",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASCODE
        ON ASCODE(NLSSORT("CODENAME",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASCOMPANY
        ON ASCOMPANY(NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("COMPANYGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_2_ASCOMPANY
        ON ASCOMPANY(NLSSORT("CLIENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_3_ASCOMPANY
        ON ASCOMPANY(NLSSORT("COMPANYGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASDISBURSEMENT
        ON
ASDISBURSEMENT(NLSSORT("DISBURSEMENTSTATUSCODE",'nls_sort=''BINARY_CI'''),

```

```

NLSSORT("POLICYGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("ACTIVITYGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("DISBURSEMENTGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("ROLEGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_2_ASDISBURSEMENT
    ON ASDISBURSEMENT(NLSSORT("POLICYGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASDISBURSEMENTAPPROVAL
    ON
ASDISBURSEMENTAPPROVAL(NLSSORT("DISBURSEMENTAPPROVALGUID",'nls_sort=''BINARY_
CI'''))
GO

CREATE INDEX CI_2_ASDISBURSEMENTAPPROVAL
    ON
ASDISBURSEMENTAPPROVAL(NLSSORT("DISBURSEMENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_3_ASDISBURSEMENTAPPROVAL
    ON ASDISBURSEMENTAPPROVAL(NLSSORT("STATUSCODE",'nls_sort=''BINARY_CI'''),
NLSSORT("DISBURSEMENTGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_2_ASDISBURSEMENTFIELD
    ON ASDISBURSEMENTFIELD(NLSSORT("FIELDNAME",'nls_sort=''BINARY_CI'''),
NLSSORT("DISBURSEMENTGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("TEXTVALUE",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_1_ASPOLICY
    ON ASPOLICY(NLSSORT("COMPANYGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("PLANGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("POLICYNUMBER",'nls_sort=''BINARY_CI'''),
NLSSORT("STATUSCODE",'nls_sort=''BINARY_CI'''),
NLSSORT("POLICYGUID",'nls_sort=''BINARY_CI'''))
GO

CREATE INDEX CI_2_ASPOLICY
    ON ASPOLICY(NLSSORT("COMPANYGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("POLICYGUID",'nls_sort=''BINARY_CI'''))

```

GO

```
CREATE INDEX CI_3_ASPOLICY
    ON ASPOLICY(NLSSORT("POLICYGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_1_ASROLE
    ON ASROLE(NLSSORT("STATUSCODE", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ROLECODE", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("POLICYGUID", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("CLIENTGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_2_ASROLE
    ON ASROLE(NLSSORT("ROLECODE", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("COMPANYGUID", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("CLIENTGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_3_ASROLE
    ON ASROLE(NLSSORT("CLIENTGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_1_ASROLEFIELD
    ON ASROLEFIELD(NLSSORT("FIELDNAME", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("ROLEGUID", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("TEXTVALUE", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_1_ASSUSPENSE
    ON ASSUSPENSE(NLSSORT("SUSPENSEGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_1_ASSUSPENSEFIELD
    ON ASSUSPENSEFIELD(NLSSORT("SUSPENSEGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_2_ASSUSPENSEFIELD
    ON ASSUSPENSEFIELD(NLSSORT("FIELDNAME", 'nls_sort=' 'BINARY_CI' ' '),
NLSSORT("SUSPENSEGUID", 'nls_sort=' 'BINARY_CI' ' '))
GO
```

```
CREATE INDEX CI_1_ASTRANSACTION
```

```
        ON ASTRANSACTION(NLSSORT("TRANSACTIONGUID",'nls_sort=''BINARY_CI'''))
GO
```

```
CREATE INDEX CI_2_ASTRANSACTION
        ON ASTRANSACTION(NLSSORT("TRANSACTIONGUID",'nls_sort=''BINARY_CI'''),
NLSSORT("TRANSACTIONNAME",'nls_sort=''BINARY_CI'''))
GO
```

```
CREATE INDEX CI_3_ASTRANSACTION
        ON ASTRANSACTION(NLSSORT("TRANSACTIONNAME",'nls_sort=''BINARY_CI'''),
NLSSORT("TRANSACTIONGUID",'nls_sort=''BINARY_CI'''))
GO
```

## APPENDIX C: WEB SERVICE SECURITY SCRIPTS

There are two Web Service security scripts you may need to run. The following scripts are important because they grant each primary company access to the Web Services. They are as follows:

1. Run the first script if you add a new company or delete a company.
2. Run the second script if no web services appear to be available. You may need to repopulate the AsAuthWebService table.

**Note:** If you cannot access a Web Service after reloading security scripts, re-run your Web Service definitions to make sure the Web Services you are referencing in the security scripts are the same ones in your database.

## SQL SERVER DATABASE SCRIPTS

1. Execute the following SQL Commands to update AsAuthCompanyWebService.

```
DELETE FROM AsAuthCompanyWebService

INSERT INTO AsAuthCompanyWebService
SELECT AuthCompanyGUID, AuthWebService.AuthWebServiceGUID
FROM AsAuthCompany,
    (SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID
    UNION ALL
    SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID
    UNION ALL
    SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID
    UNION ALL
    SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID)
AuthWebService
GO
```

2. Execute the following SQL Commands to update AsAuthWebService.

```
INSERT INTO AsAuthWebService (AuthWebServiceGUID, WebServiceName)
SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID,
'FileReceived'
AS WebServiceName
    UNION ALL
SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID,
'InputRequest'
AS WebServiceName
    UNION ALL
SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID,
'DebuggerService'
AS WebServiceName
    UNION ALL
SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID,
'ExposedComputation'
AS WebServiceName
GO
```

## ORACLE DATABASE SCRIPTS

1. Execute the following SQL Commands to update AsAuthCompanyWebService.

DELETE FROM AsAuthCompanyWebService

```
INSERT INTO AsAuthCompanyWebService
SELECT AuthCompanyGUID, AuthWebService.AuthWebServiceGUID
FROM AsAuthCompany,
    (SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID
FROM DUAL
    UNION ALL
    SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID
FROM DUAL
    UNION ALL
    SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID
FROM DUAL
    UNION ALL
    SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID
FROM DUAL)
AuthWebService
GO
```

2. Execute the following SQL Commands to update AsAuthWebService.

```
INSERT INTO AsAuthWebService (AuthWebServiceGUID, WebServiceName)
SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID,
'FileReceived'
AS WebServiceName FROM DUAL
    UNION ALL
SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID,
'InputRequest'
AS WebServiceName FROM DUAL
    UNION ALL
SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID,
'DebuggerService'
AS WebServiceName FROM DUAL
    UNION ALL
SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID,
'ExposedComputation'
AS WebServiceName FROM DUAL
GO
```

## DB2 DATABASE SCRIPTS

1. Execute the following SQL Commands to update AsAuthCompanyWebService.

```
DELETE FROM AsAuthCompanyWebService

INSERT INTO AsAuthCompanyWebService
SELECT AuthCompanyGUID, AuthWebService.AuthWebServiceGUID
FROM AsAuthCompany,
    (SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID
FROM SYSIBM.SYSDUMMY1
    UNION ALL
    SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID
FROM SYSIBM.SYSDUMMY1
    UNION ALL
    SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID
FROM SYSIBM.SYSDUMMY1
    UNION ALL
    SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID
FROM SYSIBM.SYSDUMMY1 )
AuthWebService
GO
```

2. Execute the following SQL Commands to update AsAuthWebService.

```
INSERT INTO AsAuthWebService (AuthWebServiceGUID, WebServiceName)
SELECT '7B629464-31DE-4A2C-B415-F9BD45F492FA' AS AuthWebServiceGUID,
'FileReceived'
AS WebServiceName FROM SYSIBM.SYSDUMMY1
    UNION ALL
SELECT '5ADC18E4-D752-4D3D-BEFE-5AA626210768' AS AuthWebServiceGUID,
'InputRequest'
AS WebServiceName FROM SYSIBM.SYSDUMMY1
    UNION ALL
SELECT '5007146B-326D-447C-B11B-F1A9CD7489B2' AS AuthWebServiceGUID,
'DebuggerService'
AS WebServiceName FROM SYSIBM.SYSDUMMY1
    UNION ALL
SELECT 'FD913858-B77A-40B1-9B6A-71E1191AA807' AS AuthWebServiceGUID,
'ExposedComputation'
AS WebServiceName FROM SYSIBM.SYSDUMMY1
GO
```