



Oracle® Technology Whitepaper

Oracle® E-Business Suite11i (Applications)
Oracle 10g RAC & CRS Configuration Using AutoConfig

Niklas Iveslatt
Senior Consultant
2/1/2007

**Oracle E-Business Suite11i (Applications) Oracle 10g RAC &
CRS Configuration Using AutoConfig**

By Niklas Iveslatt
February 2007

Copyright © 2007 Arisant LLC. All rights reserved.

Arisant LLC
4643 S Ulster St, Ste 800
Denver, CO 80237

Inquiries:
Tel: 303.330.4065
Fax: 888.889.0155
www.arisant.com

Oracle, JD Edwards, and PeopleSoft are registered trademarks of
Oracle and/or its affiliates. Other names may be trademarks of their
respective owners.

Abstract

This whitepaper describes the process for enabling AutoConfig with Oracle 10g RAC and CRS. If this is not done correctly, the advantages of running Oracle RAC are diminished.

Contents

Introduction	5
Configuration	6
Steps	7
1. Enable AutoConfig and Create Applications Context File.....	7
2. Run AutoConfig	9
3. Configure Listener and Update OCR	11
4. Modify RAC Specific Database and Instance Parameters	12
Metalink References	13

Introduction

In order for E-Business Suite to fully utilize RAC and to minimize manual configuration changes, it is imperative to ensure that AutoConfig is utilized and configured properly to work with Oracle RAC and Oracle CRS. Otherwise, the E-Business Suite Tiers might not use the RAC cluster fully, and some of the features such as scalability and failover might not work properly. Unless certain configuration settings are reviewed and the RAC database monitored, this problem could go un-noticed until it is too late, such as at the time of a severe RAC node failure, or at month or year end when resources are more utilized; these are not advantageous times to be changing configuration settings.

This document will provide guidance on how to configure E-Business Suite 11i in conjunction with Oracle CRS and Oracle RAC.

Configuration

The Applications architecture is a [split database tier](#) with no Applications specific components running on the database tier. The specific configuration environment utilized is depicted in this table.

System Component	Configuration	Comment
Database Nodes	2 (telcodb01,telcodb02)	
Database O/S	RedHat AS 4.0 U3	
Database Version	10.2.0.2	
Clusterware (CRS) Version	10.2.0.2	
DB ORACLE_HOME	/opt/app/oracle/product/10.2.0/prod_1	Applications specific Oracle Home utilized.
Application Tier Nodes	1 (apprd)	
Application Tier O/S	RedHat AS 4.0 U3	
E-Business Suite Version	11.5.10.2	

It is important to point out that the CRS will not function properly unless all Oracle Homes are owned by the same O/S user as the CRS home. This may not be typical in most non-RAC Applications environments. In this document, a separate Oracle Home was used for the particular Applications environment as to facilitate a cleaner cloning process of the database tier.

The process will be documented as a step-by-step process where some steps may or may not be applicable to all environments depending on the current state.

Note: This document does not cover the migration of a single database to a RAC node cluster. This is sufficiently covered in other documents such as Metalink Note: 362135.1 Configuring Oracle Applications Release 11i with 10g R2 RAC and ASM.

Steps

1. Enable AutoConfig and Create Applications Context File

If AutoConfig is not installed on the database tier, the first thing to do is to install it. This will allow for AutoConfig to run to register and configure the database node, configure related services and most importantly discover that it is part of a RAC cluster.

If multiple databases supporting different Applications environments are to co-exist in the RAC cluster, each one needs a virtual hostname. This should not be confused with the virtual IPs that were configured when Oracle CRS was installed and configured. These virtual hostnames are needed because AutoConfig uses this hostname as part of the listener name when generating the listener.ora file. This is needed to work around a limitation in Oracle CRS where listener names need to be unique.

For usability, the following format of the virtual hostnames is suggested:

```
<Database Name>_<HOSTNAME>. e.g. prod_telcodb01
```

The reason why the hostname needs to be suffixed with the real hostname `_telcodb01` will be clear later. It has to do with how `netca` works in an Oracle RAC environment. This virtual hostname does not need to be added to DNS, adding it to the hosts file is sufficient since it will only be used for AutoConfig node specific processing.

Initially, the Applications Utilities and AutoConfig install image must be created at one of the Application Tiers. These are the steps:

1. Logon to the Applications Tier as the Applications Manager (e.g. `applmgr`).
2. Execute: `perl admkappsutil.pl`. This will generate the Applications Utilities installation image.
3. Change directory to `$APPL_TOP/admin/out`.
4. Copy the file `appsutil.zip` to all Database Tiers Oracle Homes, e.g.

```
scp appsutil.zip  
telcodb01:/u01/app/oracle/product/10.2.0/prod_1.
```

5. Logon to each Database Tier node and change directory to `$ORACLE_HOME`.
6. Execute `unzip -o appsutil.zip` to perform the actual installation.

Now that the Applications Utilities and AutoConfig are installed, it is time to generate the Applications Context file. These are the steps:

1. Cleanup non-RAC context information. Logon to `sqlplus` as the Applications user and run:

```
exec fnd_conc _clone.setup_clean;
commit;
```
2. Change directory to `$ORACLE_HOME/appsutil/bin`.
3. Execute:

```
perl adbldxml.pl tier=db
appsuser=<Applications user >
appspasswd=<Applications user password>
```

When `adbldxml.pl` executes, it prompts for a number of questions. The first question is the hostname. If running the database in a RAC cluster supporting multiple Applications environments, the virtual hostname needs to be entered (e.g. `prod_telcodb01`). The second question is the port number. This needs to be correct port number associated with the node pool chosen for the Applications environment. (e.g. 1523 which is associated with node pool 2.). For reference, an Applications node pool calculator can be found in Metalink note [216664.1 FAQ: Cloning Oracle Applications Release 11i](#). The third question is the instance name and this needs to be the instance name present on the local RAC node, e.g. (`prod1` assuming the database name is `prod`.)

It is important that a listener is available, listening on port 1523 on the virtual hostname `prod_telcodb01` hostname. This can be a temporary listener setup specifically to support the connection from `adbldxml.pl`. This listener is not needed for any other purpose. This is because `adbldxml.pl` needs to connect to the local RAC database as part of the context file generation.

After successful completion of the execution, the context (`.xml`) file be generated in the `$ORACLE_HOME/appsutil` directory.

2. Run AutoConfig

With the context file available from Step 1, it is now time to prepare for the execution of AutoConfig. The current version of AutoConfig used is not capable of picking up the CRS virtual hostnames, e.g. `telcodb01-vip`. This is very important since these are the addresses that need to be listened on. (Please do not confuse this with the Applications environment specific virtual hostname.) Thus the context file needs to be updated. Edit the context file, find the key `s_virtualhost_name` and set it to the CRS virtual hostname, e.g. `telcodb01-vip` (no domain). This key will be used by AutoConfig to generate the RAC specific connection information in files such as `listener.ora` and `tnsnames.ora`.

1. Change directory to `$ORACLE_HOME/appsutil/bin`.
2. Execute `perl adconfig.sh contextfile=<Context File> appspass=<Applications user password>`

Note: Before actually running AutoConfig, it is a good idea to run `adchkcfg.sh`, which will highlight which changes will be made without actually making them. Performing this step can save from troubleshooting issues later on.

AutoConfig will read the context file and make numerous database (Net Services Topology Data Model) and file system updates. Upon completion, there will among other changes be new network files in `$ORACLE_HOME/network/admin/${ORACLE_SID}_<HOSTNAME>/`. There will also be an environment (`.env`) file located in `$ORACLE_HOME`. This will setup the environment with the proper configuration settings (e.g. `TNS_ADMIN`, `CONTEXT_NAME`, etc.). This environment file should be sourced when working in the particular Applications Environment. It might be a good idea to add the following piece of code to the `oraenv` custom section:

```
if [ -f ${ORACLE_HOME}/${ORACLE_SID}_`hostname -s`.env ]; then
    . ${ORACLE_HOME}/${ORACLE_SID}_`hostname -s`.env
else
    unset TNS_ADMIN
fi
```

This will ensure that all environment variables are set correctly when switching system identifiers (SIDs) including Applications environment and non-Applications environments.

At this point, AutoConfig needs to be run on the second node and then more importantly, it needs to be re-run on the first node, because the first node did not know about the second node when it first executed. This will ensure that `tnsnames.ora` and `listener.ora` contain both nodes' CRS virtual IPs.

3. Configure Listener and Update OCR

After running Step 2, there is a listener created but it is not registered in CRS. Registering it in CRS is desirable since CRS is responsible for the management of all clustered services and it determines the time to startup and shutdown such services.

The only supported way of registering a listener in CRS is using `netca` and this means creating a clustered listener called `listener_<ORACLE_SID >`. `Netca` will automatically add on the hostname, e.g. `listener_prod_telcodb01`. Recall in Step 1 how the virtual hostname was added; this `netca` created listener will have the same name as the one created by AutoConfig and will thus be compatible.

Note: Make sure `TNS_ADMIN` is not set while running `netca`. If this is set, the current version of `netca` will copy the corresponding files to incorrect locations on the remote RAC nodes because it will consider `TNS_ADMIN`.

At this point, there is a new listener created and the corresponding listener file is located in `$ORACLE_HOME/network/admin`, which is not the correct location for an Applications environment. In addition, this listener does not have all the configuration settings needed by Applications. This is fixed by the following process:

1. Kill the new listener on all the nodes
2. Rename the `listener.ora` and `sqlnet.ora` files located in `$ORACLE_HOME/network/admin` to `.old` files. Otherwise, they might be picked up by accident.
3. Start the applications listener, e.g. `srvctl start listener -n telcodb01 -l "LISTENER_PROD_TELCODB01"`. Repeat for all nodes.

In addition, in order to make sure that CRS will start the instances with the right environment set, execute the following command on all nodes (substituting the instance identifier of course):

```
srvctl setenv instance -d PROD-i PROD1 -t
"TNS_ADMIN=/opt/app/oracle/product/10.2.0/prod_1/network/admin/PROD1_
telcodb01"
```

4. Modify RAC Specific Database and Instance Parameters

AutoConfig generates two TNS entries that need to be applied to the database and instances: PROD_REMOTE. The parameter remote_listener needs to be updated with this value. This parameter will allow the instances to register themselves with remote node listener to allow for server side listener load balancing and failover. To update this parameter:

```
alter system set remote_listener=PROD_REMOTE
```

In addition, one instance specific parameter needs to be updated: local_listener. This parameter will allow each instance to register itself with the local node listener. To update this parameter:

```
alter system set local_listener = PROD1_LOCAL sid='PROD1'  
and  
alter system set local_listener = PROD2_LOCAL sid='PROD2'.
```

At this point it is a good idea to bounce the instances and the listeners to ensure that they are started with in the correct environment context.

Metalink References

Note	Link and Description
165195.1	Using AutoConfig to Manage System Configurations with Oracle Applications 11i
260887.1	Steps to Clean Nonexistent Nodes or IP Addresses from FND_NODES