



***Oracle EBS 12.2 Automated Clone Implementation Procedures
Candid Source LLC
May 9, 2024***

Contact for any questions or customizations – support@candidsource.com

Note:

Scripts provided is owned by Candid Source LLC and cannot be distributed or resale.

Please DO NOT Execute the scripts in PRODUCTION environment.

This is free version and Candid Source LLC is not liable for any issues.

These scripts should be tested in non-production instances that do not have any business impact.

Candid Source LLC

10120 S Eastern Ave, STE 200, Henderson, NV 89052. Ph: 1-877-999-5810. Fax: 1-877-999-5840

www.CandidSource.com

About this document:

This document describes the Oracle EBS 12.2 automated cloning procedure (EBSCloneAuto) using the scripts provided by Candid Source LLC.

The scripts use the supported oracle tools provided by Oracle and follows the Oracle Support Document – Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone (Doc ID 1383621.1)

Besides the above document, the place holder scripts are provided to cover all the specific customizations, post clone steps and Integrations.

Why EBSCloneAuto ?:

The current cloning procedure followed by clients uses the Rapid Clone tools and the steps are performed manually.

Average time taken to clone is around 24 to 48 hours and involves multiple DBAs.

Works on-prem and OCI and does not require any additional installation requirements or licenses.

No additional hardware or setup needed.

EBSCloneAuto covers and handles the following:

RAC to RAC Cloning

RAC to Non-RAC Cloning

ASM and Non ASM Storage

Fully automated clone – Database, RAC Services, AUtoconfig, Applications Tier(s) cloning, Post Clone Steps, Custom Steps, Integrations, etc

EBSCloneAuto Options:

Full Clone: This clones DB and Full Applications Stack.

Fast Clone: Coming soon as part of next release.



Technical Details:

Environment – DEV

PDBNAME - DEV

CDBNAME – DEVCDB1/DEVCDB2 (RAC)

Contains SOA Integrated Gateway.

Script Location - /u01/candidsource/DEV -- This is example and you can place in any location

EBSCloneAuto (Full Clone) consists of following scripts and is staged in directory - /u01/candidsource/DEV

Main Executables:

- For DB Tier Clone: /u01/candidsource/DEV/bin/clone_db.bin
- For App Tier Clone: /u01/candidsource/DEV/bin/clone_apps.bin



DB Tier Cloning:

Clone_db.bin performs the following:

- Shutdown Destination Database
- Cleans up ASM Directories for the Destination Database – Handled by script DEV_reset_asm.sh called by primary_db.bin
- Creates and Merges the Wallet for TDE as required for databases on OCI – Handled by script primary_db.bin
- Duplicates Destination database using RMAN active Duplicate from Standby Database – Handled by scripts DEV_startup.sql, DEV_rman_duplicate_main.sh and dup_DEVCDB_from_STBYCDB.sh. The scripts are called by primary_db.bin
- Add to the cluster, starts up the RAC Database, renames PDB and recreates services and Autoconfig – Handled by scripts DEV_startup.sql, initDEVCDB.ora.postdup, DEV_open.sql, DEV_open_post.sql called from primary_db.bin
- Add secondary node database to RAC and Configuration and Autoconfig – Handled by script DEV_pdb_open.sql and secondary_db.bin
- Cleans FND_NODES – Handled by script DEV_db_script1.sql
- Custom Script to be modified by Client – Handled by script DEV_db_script2.sql
- Custom Script to be modified by Client – Handled by script DEV_db_script3.sql
- Updates Workflow Notification Status – Handled by script DEV_db_script4.sql
- Custom Script to be modified by Client – Handled by script DEV_db_script5.sql
- Custom Script to be modified by Client – Handled by script DEV_db_script6.sql
- setUTLFileDir, createDirObject, SynUtilFileDir and Autoconfig – Handled by autocfg_db.bin
- Password changes for Non EBS apps Database Users – Handled by dba_db_script6.sql



Usage and Instructions to run clone_db.bin :

On db node 1 as oracle user,

Shutdown the database for example DEV– `srvctl stop database -db DEVCDB`

`cd /u01/candisource/DEV/bin`

execute clone_db.bin using `nohup - nohup ./clone_db.bin DEV > clone_db_DEV.log 2>&1 &`

Review the log for any failures and take action accordingly.



Glossary of scripts and files:

- clone_db.bin – Master script for Database Tier Clone
- primary_db.bin – Called from clone_db.bin. Clones Primary Instance on node 1
- secondary_db.bin – Called from clone_db.bin. Clones Secondary Instance on node 2
- autocfg_db.bin – Called from clone_db.bin. Handles UTL File Directories and runs Autoconfig on both nodes.
- run.env – Contains the environment details for Source and Target Environment and is called in several scripts to set the environment.
- DEV.env – Created by run.env and is called in several scripts to set the environment.
- DEV_reset_asm.sh – deletes the DEVCDDB database files from ASM. Creates the password file orapwDEVCDDB in ASM for DEVCDDB
- DEV_rman_duplicate_main.sh – RMAN duplicate of DEVCDDB from STBYCDDB. Calls the script dup_DEVCDDB_from_STBYCDDB.sh
- dup_DEVCDDB_from_STBYCDDB.sh – Called from DEV_rman_duplicate_main.sh to execute the RMAN duplicate.
- initDEVCDDB.ora.dup – init parameters to start up the database for duplicate. Called from sql DEV_startup.sql.
- DEV_startup.sql – This is to startup the database with no RAC and for database duplicate. This also handles the TDE wallet creation.
- initDEVCDDB.ora.postdup – This script is called after duplicate by primary_db.bin. This enables cluster_database=true
- DEV_open.sql – Called from primary_db.bin. Deletes MISPRD1 Services, renames pluggable database to MISDEV and recreates all the DB Services
- DEV_open_post.sql – Called from primary_db.bin. Create spfile and startups using spfile.
- DEV_pdb_open.sql – Called from secondary_db.bin. Starts the services on second db node.



Mapping of custom post clone Sql Scripts and Place holder for future use:

DEV_db_script1.sql	Example - Cleaning FND Nodes	Execute as apps user
DEV_db_script2.sql	Place holder for your post clone script	Execute as apps user
DEV_db_script3.sql	Place holder for your post clone script	Execute as apps user
DEV_db_script4.sql	Example - Update wf_notifications table	Execute as apps user
DEV_db_script5.sql	Example - Recreate Database Links	Execute as apps user
DEV_db_script6.sql	Example - Recreate Informatica ACLs	Execute as apps user
DEV_db_script7.sql	Placeholder for your post clone script	Execute as apps user
DEV_db_script8.sql	Placeholder for your post clone script	Execute as apps user
DEV_db_script9.sql	Placeholder for your post clone script	Execute as apps user
DEV_db_script10.sql	Placeholder for your post clone script	Execute as apps user

DEV_db_script11.sql	Password changes for non EBS Database users	Execute as sys user as sysdba at PDB level
DEV_db_script12.sql	Placeholder for your post clone script	Execute as sys user as sysdba at PDB level
DEV_db_script13.sql	Placeholder for your post clone script	Execute as sys user as sysdba at PDB level
DEV_db_script14.sql	Placeholder for your post clone script	Execute as sys user as sysdba at PDB Level



ENV files:

/home/oracle/.bash_DEV – this will set . \$ORACLE_HOME/DEV_<node1>.env

/home/oracle/DEV_pdb – This will set ORACLE_PDB_SID=DEV to execute the sql as sysdba user at PDB Level

The above env files are called from various scripts.

Add the following Tns Entry in Oracle Home-

```
DEV_CLONE=
(DESCRIPTION=
  (LOAD_BALANCE=YES)
  (FAILOVER=YES)
  (ADDRESS=(PROTOCOL=tcp)(HOST=<host>-scan.<Domain>)(PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=ebs_DEV)
  )
)
```

The above tns entry is used by post clone sql scripts to connect to PDB as apps or system user



Apps Tier Cloning:

Make sure to stage the pairs file as follows:

Primary apps node 1 pairs file as dev_pairs_fs1

Secondary apps node 2 pairs file as dev_pairs_add_fs1

Clone_apps.bin performs the following -

- Creates apps run fs – Moves current fs to recycle_bin. Creates source run fs. This is handled by script create_fs.sh
- Runs adcfgclone on primary node – Handled by script primary_apps.bin
- Adds connection filter for second apps node on run fs – Handled by script ha_apps.bin
- Modifies config.xml to allow instead of deny – Handled by script ha_apps.bin
- Runs pre-clone on run fs of primary apps node – Handled by script ha_apps.bin
- Starts patch admin server on primary apps node – Handled by script ha_apps.bin
- Adds connection filter for second apps node on patch fs – Handled by script ha_apps.bin
- Runs pre-clone on patch fs of primary apps node – Handled by script ha_apps.bin
- Stops and starts admin server for both run and patch fs – Handled by script ha_apps.bin
- Runs adcfgclone addnode on secondary apps tier – Handled by secondary_apps.bin
- Updates data source password in weblogic to be modified by Client– Handled by chane_apps_passwords.sh
- Changes SYSADMIN, ASADMIN passwords using FNDCPASS to be modied by Client – Handled by change_apps_passwords.sh
- Runs post clone steps by copying the required BMCD Custom files on run fs to be modified by Client – Handled by script post_clone_run_custom_files.sh
- Runs autoconfig on all apps nodes – Handled by autocfg_apps.bin
- Starts Apps services – Handled by start_apps.sh
- Runs post clone steps by copying the required Custom files on patch fs to be modified by Client – Handled by script post_clone_patch_custom_files.sh
- Place holder post clone script to be modified by Client – Handed by script DEV_apps_script1.sql
- Place holder post clone script to be modified by Client – Handled by script DEV_apps_script2.sql
- Place holder post clone script to be modified by Client – Handled by script DEV_apps_script3.sql
- Place holder post clone script to be modified by Client – Handled by script DEV_apps_script4.sql
- Place holder post clone script to be modified by Client – Handled by script DEV_apps_script5.sql
- Place holder post clone script to be modified by Client – Handled by script DEV_apps_script6.sql



Mapping of custom post clone Sql Scripts and Place holder for future use:

DEV_apps_script1.sql	Example - Updated Workflow Components	Execute as apps user
DEV_apps_script2.sql	Example - Table and Profile Updates	Execute as apps user
DEV_apps_script3.sql	Example -Modifies Oracle DBA Directories	Execute as apps user
DEV_apps_script4.sql	Placeholder Script	Execute as apps user
DEV_apps_script5.sql	Placeholder Script	Execute as apps user
DEV_db_script6.sql	Placeholder Script	Execute as apps user
DEV_db_script7.sql	Placeholder for future use	Execute as apps user
DEV_db_script8.sql	Placeholder for future use	Execute as apps user
DEV_db_script9.sql	Placeholder for future use	Execute as apps user
DEV_db_script10.sql	Placeholder for future use	Execute as apps user



Usage and Instructions to run clone_apps.bin :

On apps <Apps Server > as applmgr user,

cd /u01/candidsource/DEV/bin

execute clone_apps.bin using nohup - nohup ./clone_apps.bin DEV > clone_apps_DEV.log 2>&1 &

Once the db clone process is complete, log file will be emailed for verification for any errors.

Review the log for any failures and take action accordingly.



Glossary of scripts and files:

- clone_apps.bin – Master script for Apps Tier Cloning
- primary_apps.bin – Clones primary apps node. This is called by clone_apps.bin
- secondary_apps.bin – Clones secondary apps node. This is called by clone_apps.bin
- ha_apps.bin – This adds connection filters, modifies config.xml, runs pre-clone of primary apps node on both run/patch fs, stops and starts admin server on both run/patch fs. This is called by clone_apps.bin
- autocfg_apps.bin – Runs autoconfig on both apps tier nodes. This is called by clone_apps.bin
- post_clone_run_custom_files.sh – This copies all the custom files to run_fs. This is called by clone_apps.bin
- post_clone_patch_custom_files.sh – This copies all the custom files to patch fs. This is called by clone_apps.bin
- start_apps.sh – Starts all application services

