

Author – A.Kishore/Sachin
<http://appsdba.info>

VARIOUS MODES/OPTIONS OF ADPATCH

There are various options available for applying patch in Oracle apps 11i and R12 environment. ADpatch comes with lots of option that can be used, especially when we are applying the patch in production.

Modes of ADPATCH

There are 3 modes of applying a patch.They are as follows :

- 1) Pre-Install Mode
- 2) Test Mode
- 3) Non-Interactive Mode

We will know all modes one by one in detail. Let's start.

1) Pre-Install Mode

Pre-install mode is used to update AD utilities before an upgrade and to apply family consolidated upgrade packs.AutoPatch Pre-AutoInstall mode allows you to apply patches when your installation is missing database information and/or filesystem information that AutoPatch requires to run in normal mode.

Examples of when we run AutoPatch in Pre-AutoInstall mode (and cannot run it in normal mode) includes:

- Prior to installing Oracle Applications for the first time
- Prior to upgrading Oracle Applications to the latest release.
- During an upgrade (to apply a bug fix that stopped your upgrade)

Applying patch in pre-install mode performs following tasks:

- Version checking
- File copy actions
- Relink FND and AD executables
- Save patch history information to file system

AutoPatch in pre-install mode will NOT:

- Run SQL of EXEC command
- Generate files
- Read product driver files
- Apply maintenance pack

To apply patch in pre-install mode, run `adpatch preinstall=y`

Author – A.Kishore/Sachin
<http://appsdba.info>

2) Test Mode

AutoPatch provides a test mode in which it tells us everything it would have done in applying a patch, but doesn't actually apply the patch.

To run AutoPatch in Test Mode, we must include 'apply=no' on the AutoPatch command line. For example:

```
$ adpatch apply=no
```

Instead of performing an action, AutoPatch indicates that it is not performing the action because "Apply=No". In general, AutoPatch lists each file it would have copied, generated, relinked, or executed. This shows you exactly what actions it would have performed.

AutoPatch test mode works the same as normal mode, with the following exceptions:

- It does not copy any files from your patch directory into your installation area.
- It does not copy any files from your APPL_TOP to JAVA_TOP or OAH_TOP.
- It does not archive any object modules into your product libraries.
- It does not generate any forms or reports.
- It does not relink any executables.
- It does not run any 'sql' or 'exec' commands.
- It does not update the release version in the database.
- It does not update the patch history file.

AutoPatch asks us the same initial questions in test mode as in normal mode. It performs the following actions to determine what it would have done if run in normal mode:

- Reads and validates the patch driver file.
- Reads product file driver files.
- Extracts object modules from your product libraries (so it can perform version checking on the object modules it extracts).
- Performs version checking.
- Looks in the database to determine what 'sql' and 'exec' commands it would have run.

It's always good to run the patch in test mode and analyze the things before applying the patch in normal mode.

Author – A.Kishore/Sachin
<http://appsdba.info>

3) Non-Interactive Mode

Starting in Release 11.5, we can run AutoPatch non-interactively.

Before we can run AutoPatch non-interactively, for this mode we must first create an AutoPatch defaults file for our current environment.

Here are the simple steps to create an AutoPatch defaults file for our current environment:

1. Specify defaultsfile=<New Defaults File Name> on the AutoPatch command line. The defaults file must be located under \$APPL_TOP/admin/<SID>.

For example:

```
adpatch defaultsfile=$APPL_TOP/admin/testdb1/my_def.txt
```

2. Run AutoPatch up to the point where it asks you for the directory where your Oracle Applications patch has been unloaded. Then type 'abort' at this prompt.
3. Verify that our defaults file exists.

Once we have an AutoPatch defaults file for our current environment, you can run AutoPatch non-interactively.

Before applying any Oracle Applications patch, either interactively or non-interactively, we should read the README file (usually called readme.txt) supplied with the patch. We should also read the documentation supplied with the patch (if any).

It is possible to apply just a single patch driver file non-interactively using AutoPatch. Here is an example:

- defaults file is \$APPL_TOP/admin/PROD/def.txt
- Applying copy driver for patch 123456, which is located in directory \$APPL_TOP/patch/123456.
- Using three parallel workers
- AutoPatch log file name is cpy123456.log

Author – A.Kishore/Sachin
<http://appsdba.info>

The AutoPatch command line would be:

```
adpatch defaultsfile=$APPL_TOP/admin/PROD/def.txt \  
logfile=cpy123456.log \  
patchtop=$APPL_TOP/patch/123456 \  
driver=c123456.drv \  
workers=3 \  
interactive=no
```

If we don't give any of the mode as mentioned above and apply the patch simply using adpatch command then Interactive will work which is normal mode of patch application.

Having seen the modes of patch application, now we will see various arguments for applying patch.

1) defaultsfile

Purpose: This option is used when we are running the patch in non-interactive mode. In that case we create defaults file and provide that file as an option for running patch in non-interactive mode.

Default: none. No default file read or written.

2) logfile

Purpose: This is the name of adpatch log file which it will write during patch application.

Default: none. Adpatch prompts for this value.

3) workers

Purpose: Specifies the number of workers to run. This value depends on number of CPU and other factors.

Default: none. Adpatch prompts for this value.

4) patchtop

Purpose: Top-level directory for the current patch. This is the directory after unzipping the patch. This directory will have the patch number.

Default: none. Adpatch prompts for this value.

5) driver

Purpose: Name of the patch driver file. This comes with the patch and is present in patch directory.

Default - none. Adpatch prompts for this value.

Author – A.Kishore/Sachin
<http://appsdba.info>

6) restart

Purpose: To restart an existing session. Only valid when interactive=no is specified.
Default: No

7) localworkers

Purpose: Used in Distributed AD to specify the number of workers to be run on the current machine. If you have multi node instance (example RAC and shared APPL_TOP), then you can utilize this paramter to run the patch parallely in multiple nodes. You can start few workers on node 1, few on node 2 and so on. The way this can be done is that, you can start adpatch on one node with localworker=<some value less then total workers>. Then run adctrl on other node in distributed mode and start some mode workers. This will speed up the process and utilized the resources effectively.

Default: Value specified for workers.

8) printdebug

Purpose: To display extra debugging information.
Default: No.

Now lets consider some common options that can be used with adpatch options=<value>

1) checkfile

Purpose: To skip running exec, SQL, and exectier commands if they are recorded as already run. Indicates that Autopatch should run the command *only* if a certain file is newer than the version of it that was last run. The idea behind it is to reduce the duration of an Autopatch session by skipping actions that don't really need to be performed. When used in the right manner, it can dramatically improve Autopatch performance, especially for big patches and/or long running actions.

Default: checkfile (use 'nocheckfile' to skip)

2) compiledb

Purpose: To compile invalid objects in the database after running actions in the database driver.

Default: compiledb (use 'nocompiledb' to skip)

3) compilejsp

Purpose: To compile out-of-date JSP files, if the patch has copy actions for at least one JSP file.

Default: compilejsp (use 'nocompilejsp' to skip)

Author – A.Kishore/Sachin
<http://appsdba.info>

4) copyportion

Purpose: To run commands found in a copy driver. This will copy the higher version files from patch to product top.

Default: copyportion (Use 'nocopyportion' to skip. Use it only when mentioned in readme of patch)

5) databaseportion

Purpose: To run commands found in a database driver. This portion includes applying the files (like sql, pls etc) to database.

Default: databaseportion (use 'nodatabaseportion' to skip. Use it only when mentioned in readme of patch)

6) generateportion

Purpose: To run commands found in a generate driver. This portion will generate new executable files from the copied code of patch. For example it will generate new forms files (fmx) from new .fmb files.

Default: generateportion (use 'nogenerateportion' to skip)

7) integrity

Purpose: To perform patch integrity checking. Tells adpatch whether to perform patch integrity checking, which verifies that the version of each file referenced in a copy action matches the version present in the patch.

Default: nointegrity (By default the integrity is not checked)

8) maintainmrc

Purpose: To maintain the MRC schema after running actions found in the database driver.

Default: maintainmrc (use 'nomaintainmrc' to skip)

9) autoconfig

Purpose: Tells adpatch to run Autoconfig after patch installation.

Default: autoconfig (use 'noautoconfig' to skip)

10) parallel

Purpose: To run actions that update the database or actions (like SQL) that generate files in parallel (like genform).

Default: parallel (use 'noparallel' to skip)

Author – A.Kishore/Sachin
<http://appsdba.info>

11) prereq

Purpose: Tells adpatch whether to perform prerequisite patch checking prior to running patch driver files that contain actions normally found in the copy driver.

Default: prereq (use 'noprrereq' to skip)

12) validate

Purpose: To connect to all registered Oracle Applications schemas at the start of the patch. Adpatch validates the passwords for each schema.

Default: novalidate (use 'validate' to validate schema passwords)

Following flags can be passed to adpatch

1) hidepw

Purpose: This argument is used to hide the passwords in log files

Default: nohidepw

2) trace

Purpose: Tells the adpatch utility whether to log all database operations to a trace file

Default: notrace

3) logging

Purpose: Tells the adpatch utility whether to create indexes using the logging or nologging mode.

Default: logging