

SELECT[®] Online

The Journal of the International Oracle Users Group

Select Journal
Home

Regular Columns

Past Issues

General Information

IOUG Home



Select Magazine - November 2001

Volume 8, No. 1

Oracle Applications Patching

Jonathan Vincenzo, USC

Oracle Applications patching is a critical part of maintaining an Oracle Applications environment. There are several utilities that are used to perform this task; however, improper understanding and usage of these utilities can do a lot of damage. The person ultimately responsible for using these utilities should only be the Applications System Administrator/DBA. To be successful at patching, you will need access to the Unix applications owner account, typically 'aplmgr,' because there will be functions performed on both the database and concurrent processing servers.

After receiving the request for a patch from the applications functional team, determining the need by research on Metalink or via a request to resolve a specific issue through contact with Oracle Support, the patch is downloaded via Oracle's Metalink support site to the server.

Before you apply any patches, you must first determine which patches have been applied to your current Oracle Applications installation. The file applptch.txt can be reviewed for this purpose. This file is located in \$APPL_TOP/admin/<SID>. This file MUST be present in order to apply patches.

Please note, I do *not* recommend applying patches 'to stay current.' Many patches released by Oracle have unforeseen consequences and are immediately withdrawn. Patches also have been known to break seemingly unrelated modules. This is a *caveat emptor* situation, if ever there was one. Make sure that you fully understand the purpose of a patch, and that the Oracle support tech does, too, before you attempt to apply a patch.

To find out which patches have been applied, for example, to the Accounts Receivable(ar) module, execute the following command from the Unix prompt:

```
grep 'begin bug ar' applptch.txt
```

This will output a listing of all patches that have been applied to date.

In addition to 'applptch.txt,' a separate list of patches that have been applied and the order in which they were applied also should be kept. An Excel spreadsheet that has a main sheet that contains an overall list of patch number, the module, the date they are applied and the person who applied it and separate sheets for each module installed containing the date applied, patch number and a brief explanation of the function has worked extremely well for me.

The location of all patch files should be in a consistent location across all nodes. The patch is then untarred/gzipped. Each patch will contain a 'readme' file. This file should be carefully reviewed for prerequisite patches, pre-install actions and post-install actions.

Prerequisite patches are then downloaded and the new 'readme' file is reviewed for prerequisites, pre-install and post-install steps.

Once all of the appropriate prerequisite patches have been downloaded, they are applied in the order determined from the 'readme' files.

The first step in all patching operations is to stop all of the concurrent managers, the database, and the listener and take a full cold backup of the system.

I cannot stress this enough. Make a cold backup of the system before you begin this process. If there is not enough space on disk to back up the entire \$APPL_TOP directory and the database, dump everything onto a tape. Do *not* apply patches before taking a backup of your current system.

Database drivers *cannot* be backed out without a system backup. You *may* be able to back out other drivers by reversing the steps, but there is no guarantee. If you need to back out a patch, going to a backup is the safest, simplest method. In most instances, it is the *only* method.

Individual patches contain separate drivers. Usually a 'c,' 'd' and 'g' drivers. These are the 'copy,' 'database' and 'generate' drivers, respectively, and should be applied in that order, unless the 'readme' states otherwise. Not all patches will contain all drivers, but the order in which they are applied remains the same.

In a multi-node environment, the 'c' and 'g' drivers are applied to the application node, and all drivers are applied to the database node. Regardless of the number of tiers, the process used to apply the patches is the same.

The tool that is used to apply a patch is called 'adpatch.' It is also referred to as 'autopatch.' Adpatch can be run interactively or non-interactively. Initially it is best to run this interactively. After you are comfortable with the patching process, it can be run non-interactively.

Before running adpatch, the local environment needs to be set. This is done with the APPSORA.env and the <SID>.ora scripts. After running these scripts you should have all of the appropriate environment variables set. After setting up your local environment run adpatch in interactive mode, you simply type 'adpatch.'

You will be asked for your \$APPL_TOP directory, the location in which to place the log for this session, whether or not you wish email notification of a failure and the batchsize. The \$APPL_TOP should be correct, the location of the logs should be a consistent location under \$APPL_TOP, and the batch size should be left at default.

Adpatch will then print the SID and the ORACLE_HOME that it is going to use. In an 11i environment, the ORACLE_HOME should be the location of the oracle version '8.0.6' executables. This is the default.

Next you will be asked for the SYSTEM and APPS password. After entering those passwords, the program will attempt to connect to the various schemas and, after successfully completing those steps, will ask the location where the patch you're applying is located. If you started adpatch from the directory that contains the drivers, the default will be correct. Otherwise, you must enter the full path to the location of the patch driver files.

You will then be prompted for the name of the driver you wish to apply. Driver files are in the format of 'driverpatchnumber.driv'. For example, c123456.driv, would be the 'c' driver for the patch number 123456.

If the driver file is a 'c' driver, adpatch will then proceed to copy the updated files into their new locations. If the driver file is a 'd' or a 'g' driver, you will be prompted to enter the number of workers.

'Adworkers' are operating system processes that allow adpatch to spawn concurrent jobs in order to complete jobs faster.

The adpatch process will spawn as many concurrent jobs as there are available workers to complete those jobs. If OS resources permit, there are no significant drawbacks to setting the number of available workers to the maximum of 99.

The adpatch utility will then proceed to apply appropriate patches. When this process is complete, or if you wish to monitor the progress of a particular worker process, the logs can be reviewed. These are located in \$APPL_TOP/admin/<SID>/log. They are in the format 'adworkworker#.log'. For example, adwork10.log would be the log for worker number 10.

After becoming familiar with adpatch, a significant time savings can be seen by running it in 'non-interactive' mode. This involves passing the applicable parameters into the utility via the command line.

Before running adpatch in non-interactive mode, a file must be created that contains the defaults for your environment. This is created by running the following command:

```
adpatch defaultsfile=$APPL_TOP/admin/<SID>/def.txt
```

Next, continue to run adpatch up to the point where it asks you for the directory where your patch has been unloaded. Then type 'abort' at this prompt. Now verify that you have a file, 'def.txt,' in the directory you specified.

While it is possible to apply all of the drivers in a patch at once, it is not recommended. This is because it is likely that some errors may occur. It is best to resolve those issues before moving on to the next driver file.

Below is the syntax for applying a single patch driver file:

```
adpatch defaultsfile =$APPL_TOP/admin/<SID>/def.txt logfile=123456.log
\  
patchtop=<LOCATION OF PATCH>/123456 driver=c123456.drv workers=3
```

After the successful application of the patch, these logs should be reviewed for any errors. It is important to note that many times an error may occur, but not reported by adpatch. The only indication that an error has occurred will be in the log.

The next step is usually to recompile the 'apps' schema. This is accomplished via the 'adadmin' utility.

Adadmin is the utility used in Oracle Applications maintenance. All major maintenance activities, for example compiling schemas, relinking applications files, regenerating forms file, among many others, are performed with this utility. It is a menu driven program, similar to adpatch, and most post-install actions are performed via adadmin.

To run adadmin, you simply type 'adadmin.' The initial questions are the same as those for 'adpatch' and the answers are similar.

After validation, you will be presented with a menu of three options. One will perform work on the database, one performs work on the Oracle Apps related files and the third exits the application.

To recompile the apps schema, select the first option, 'Maintain Applications Database Objects.' You are then presented with another menu that contains various database related functions. Select the second option 'Compile APPS schema(s).'

Adadmin will then validate your init.ora parameters to make sure they are within Oracle's recommended settings for an apps installation. If they are not, you are given the opportunity to exit adadmin and correct them or continue on.

You are then prompted for the number of workers you wish to use to perform this function. Testing has shown that no appreciable performance gain can be seen from setting this number higher than 50. Therefore, it is recommended that the number of workers be 50.

You will then be prompted whether to run Invoker's Rights processing in incremental mode. Accept the default of 'no.' Adadmin will then proceed to process and compile the apps schema. When this is complete, review the worker logs for any errors and proceed to any other post-install steps as documented in the 'readme.'

Another method for recompiling the apps schemas is to use the 'utlrp.sql' program that is located under the \$ORACLE_HOME/rdbms/admin directory on the database server. This program is significantly faster than using adadmin and may provide more insight into determining which objects will not recompile and why. It is important to note that in most apps instances, there are a number of objects that will not compile. This is because they reference objects that are part of modules that may not be installed on your system.

Additionally, the 'readme' may instruct you to relink the applications executables. The 'relink applications programs' option under 'adadmin' is used to build the Oracle applications executables. Even if it doesn't tell you to do so, it is recommended that you relink to be certain that the executables are the most current.

To relink the applications programs, you start adadmin as previously noted, select 'Maintain Applications Files' from the menu, then select 'relink applications programs.' The answers to the questions asked by the interface should be straightforward at this point.

After the program completes, review the logs for any errors as previously noted.

Another important program to be aware of during the application of patches, or when running any program that utilizes adworkers, is the 'adctrl' utility. This utility allows you to control the behavior of workers as they are performing their tasks. The adctrl utility allows you to determine the status of any worker, restart failed jobs or skip jobs.

This is typically used when a worker appears to be taking a long time to complete, to view the status of that worker, to restart a worker after a problem has been resolved or to tell a worker to skip a job that is failing and cannot be fixed.

The utility is started with the command 'adctrl.' You are then asked for the password to the 'appls' and 'apps' schemas. A menu of options that are self explanatory is then presented.

The key to being successful is studying the 'readme' files that come with each patch. In my opinion, this allows you to keep the single most critical item coordinated --- namely, the prerequisite patches. The technique that I use is, admittedly, 'low tech,' but it may help you.

I print out each readme file and highlight the prerequisites. Then I download and print out each prerequisite's readme and place it on the top of the pile. When I have exhausted the prerequisite list, only then do I begin patching. As I complete a patch, I date it and place it in another folder -- low tech but effective! This method also gives me a place to make notes about any problems encountered with a patch, and provides a hardcopy log of which patches have been applied and when.

As you can see, the steps in patching are quite simple. The trick is to approach patching in a slow, methodical method. Complete the prerequisites, and their prerequisites, apply the patch and then perform any post-install actions. It is important to remember that you should not hesitate to contact Oracle Worldwide Support if you experience any problems. In the patching process they are your best friend. They have access to a much more detailed patch database than you do.

Finally, don't get discouraged when things don't seem to go correctly. Some packages will *never* compile, some scripts will *always* generate an error. With experience, you will learn the ins and outs of this process.

About the Author

Jonathan Vincenzo is a senior consultant at TUSC. He has been an Oracle database administrator for more than eight years and a UNIX system administrator for more than three years. In his spare time, he enjoys building small boats and spending time with his wife and two basset hounds (the kids). He can be reached at vincenzoj@tusc.com or at jon_vincenzo@boatbuilding.com.

[Download Acrobat Reader](#)

Copyright 2003 by the International Oracle Users Group