

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

## Reduce Patching Downtime

This document elaborates some tips to reduce our system downtime during the application of patches.

### **1. Consolidate Patches**

Using AD Merge Patch to merge multiple patches into a single patch that eliminates the time it takes to start a new AutoPatch session.

### **2. Apply Patches in Non-interactive Mode**

You can automate much of the patching process by applying patches in non-interactive mode. In this mode, we store the responses to the patching prompts in a defaults file, and then specify the name of this default file when we run AutoPatch.

### **3. Skip Database Tasks**

We can use AutoPatch options=`nocompiledb`, `nomaintainmrc` to skip tasks such as “Compile APPS schema” and “Maintain MRC” until all patches have been applied successfully. Normally AutoPatch automatically compiles the APPS schema and maintains the MRC schema during the patching application of standard patches.

### **4. Skip upload of Patch Information**

Normally AutoPatch uploads patch history information to the database automatically each time it successfully applies a patch. The time required for the upload may be more depending on the size of the patch. We can skip this task during the AutoPatch session and upload the patch history information later.

We will use options=`phtofile` during the downtime patching session to skip the information upload. After a successful patch, we will run AutoPatch again with the argument `uploadph=y` to upload the patch history information from the patch history files to your database. AutoPatch performs the upload and then it quits.

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

## **5. Apply patch on a Test System First**

To analyze the effects of applying a patch, we should apply it first to a test system. We will check for long-running jobs and phases, which take the longest time in the timing statistics report (\$APPL\_TOP/admin//out/adt.lst).

## **6. Implement Shared Application Tier File System**

Implementing a multi-node system with a shared application tier file system saves patching time because we apply patches only once, on the primary node.

## **7. Use Distributed AD**

Distributed AD is a parallel processing feature that reduces downtime by efficiently utilizing all the available resources on a shared application file system.

## **8.Reduce Resource Related Issues**

We should modify rollback segment sizing and temporary segment space to optimize resources during patch application.

## **10. Impact analysis**

We can perform an impact analysis on a patch, without affecting system, by running AutoPatch with the command line option apply=no.

Some patches include only file system changes. Most of the Applications patches do not include database changes.

## **11. Use AutoPatch options efficiently**

Using certain AutoPatch command line options and managing the number of workers can minimize patch application time. We can use options like norevcache, nogenform, nogenrep etc to reduce patching downtime.

## **12. Merge NLS (Translation) Patches and Apply Them During Uptime**

If we have multiple patches for multiple languages, merge all US patches into a single patch. Then, merge the NLS translation patches for each active language in our system into a single patch for each language. Apply the US patches first during downtime. Then, we can apply the merged NLS translation patches during uptime.