

Staged APPL_TOP approach - Key to achieve tolerable downtime for Oracle Apps Migrations

Nikhil Kumar
Infosys Technologies Limited

Nagarjuna Cherukuri
Navo Systems Inc.

Introduction

Organizations performing Oracle Applications migration face various challenges including how to reduce application downtime during go-live window to avoid any negative impact to the business. Most of these organizations can only tolerate being down for a short weekend maintenance window, and struggle to squeeze their migration plan in this pre-defined period.

Each migration project is unique in itself, and its go-live application downtime window varies from project to project. There are significant numbers of parameters which impact the downtime period like version of the application, size of the database, number of modules implemented, language patches requirements, quality of the hardware and infrastructure, extent of customizations in system, database upgrade requirements etc. Many organizations realized after first pilot upgrade, that application downtime window can go to days or even week time for complex migrations during go-live period. It is very critical for organizations to know the tolerable downtime window available to them which will not hurt their business user community and operations.

There are many techniques and best practices available to reduce the downtime window. Staged APPL_TOP is one of the key approaches available to achieve the tolerable downtime. It saves significant time spent on copy and generate portion of patching activities and file system changes steps, which can be performed without additional downtime, hence reduce application downtime in critical path. This paper discusses the needs and benefits, methodology, and specific tips for staged APPL_TOP approach.

What is Staged APPL_TOP

Staged APPL_TOP methodology is one of the approaches available to reduce patch down time during Oracle Applications migrations and/or patch updates. This method involves cloning of the actual production environment into different environment called staged environment. All the patches are applied on this staged environment when the production system is up and running. This staged APPL_TOP is used to both synchronize and apply DB driver to actual environment. The dbportion of the patches is applied from staged APPL_TOP to actual environment when downtime begins. Synchronization of actual APPL_TOP with staged APPL_TOP can be done while db driver is running. This saves time taken for copy and generate portion.

Methodology Approach

This staged APPL_TOP methodology approach involves the following five steps.

1. Preparing the source environment
2. Creating the staged environment
3. Applying patches to the staged environment
4. Applying patches to the production environment and Synchronize
5. Post tasks on the production environment

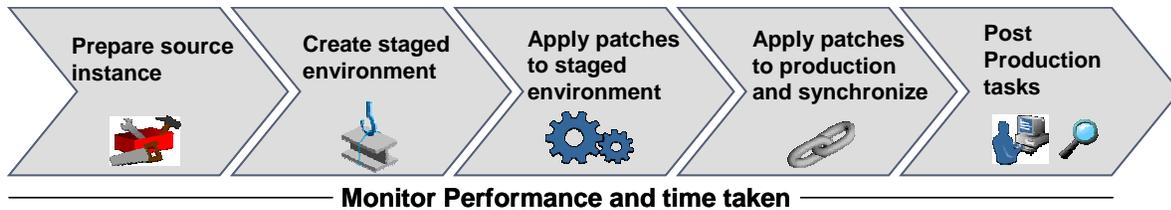


Figure 1: Staged APPL_TOP Methodology Steps

Step 1 - Preparing source environment

The source instance is referred here is the actual production environment, which needs to be migrated to higher version of Oracle. This preparation of the source instance phase involves the following steps:

1. Checking pre-requisite patches on the source environment
2. Maintaining snapshot information

Source environment should have a set of pre-requisite patches before creating the staged environment. The list of pre requisite patches can be prepared based on the following points.

- Staged APPL_TOP approach requires AD.level of both production and staged environments to be the same. So, any patches updating the AD level should be applied in production system prior to creating the staged environment.
- Staged APPL_TOP approach can not be used to apply new product patches associated with a maintenance pack. E.g. patch 2644311 updates with the new products that are not already present in the environment. So, all such patches should be applied to production environment before creating staged environment.
- Patch history export utility mentioned in step 5 is required for staged APPL_TOP. The patch #3285255 provides this patch history export utility

Maintaining snapshot is required for the right pre-requisite checking during any major patches update. So, you should update the snapshot information prior to creating or copying the production system to the staged environment. This needs to be done for each APPL_TOP in the application system.

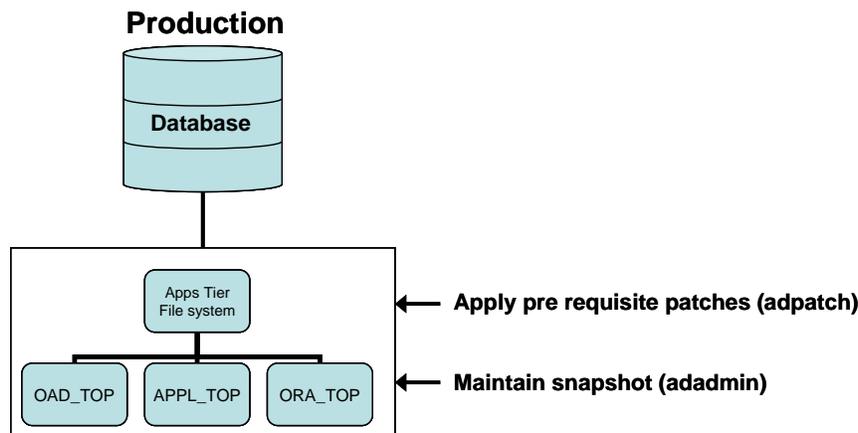


Figure 2: Preparing the Source Instance

Step 2 - Creating the staged environment

Staged environment can be created from production environment using rapidclone procedure documented in Metalink note 135792.1. However, the following points should be taken into consideration while creating the staged environment.

- The staged environment must keep the same topology as of the production environment including database, file system and all the application tiers. Each physical APPL_TOP of the production system must exist in the staged

environment. If the production system is multi-node environment, the staged environment must also be multi-node.

- Patch history is stored with respect to APPL_TOP name. So, you should have the same APPL_TOP name in both production and staged environment so that patch history can be moved from staged environment to production as mentioned in step 5.
- It is always good to have different ports and instance name for staged environment than production. This would help to avoid the updating of production database accidentally.
- There should not be any patch application to production system once staged environment is created. Snapshot information becomes invalid in staged environment with respect to production system if any patch is applied to production after creating the staged environment.

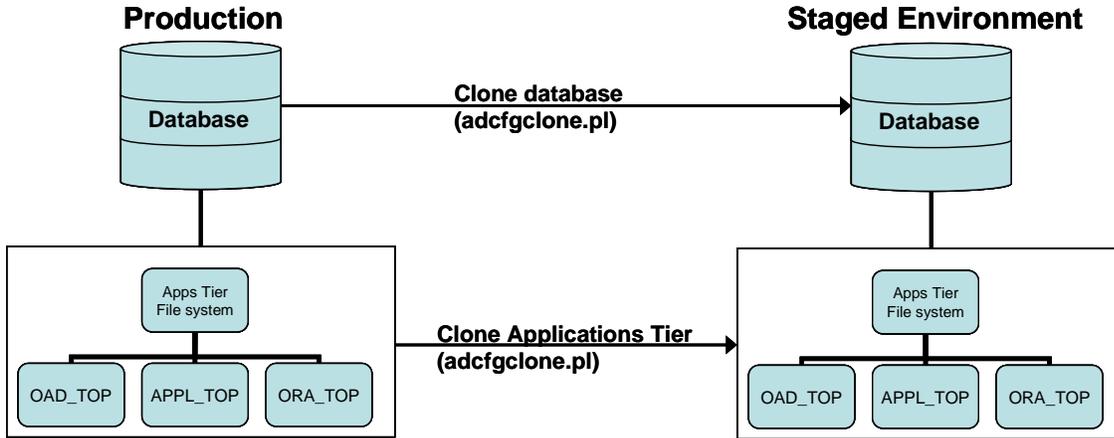


Figure 3: Creating the staged environment

Step 3 - Applying patches to staged environment

Once staged environment has been created as per the steps mentioned above, it is ready for patching. The patches can be applied in the staged environment same way you will apply patches to any other Oracle Applications system for migration requirements for required category steps. This patching is independent of actual production environment. So, it can be done while production environment is up and running.

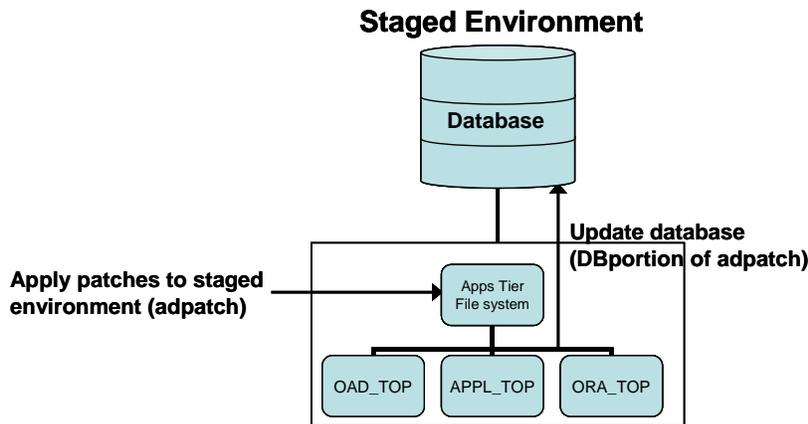


Figure 4: Applying patches to staged environment

Step 4 - Applying patches to production environment and synchronize

Production system is ready for patching by means of staged APPL_TOP once all the patches are applied to the staged environment. Actual production patching involves the following

- Checking the connectivity from staged APPL_TOP to production database
- Applying the DB portion of patches from staged APPL_TOP to production system
- Synchronizing the staged APPL_TOP with the production APPL_TOP

You need to establish database connectivity from application tier of staged environment to production database prior to actual patching. This can be achieved by adding TNS entry of production database in tnsnames.ora of 806 TNS_ADMIN of the staged environment. You need to re-point the TWO_TASK variable from staged database to the production database. This can be best done by means of .profile file sourced during UNIX login. This step can be performed while production system is up and running.

Patching to actual production environment can start once it is taken down for maintenance. Actual patching starts from the application tier of the staged environment. You need to apply the DB portion alone. Both copy and generate portion related changes are migrated by means of synchronizing the APPL_TOP and OAD_TOP. DB portion of the patching is done by applying D driver of the patches. In case of unified driver, the nocopyportion and nogenerateportion options should be used by running the adpatch with options=nocopyportion, nogenerateportion . During the patching, all the changes (.odf, .pls, .sql, .ldt etc) are applied from the files present under PROD_TOPs of the staged environment.

Synchronizing of applications tier file system can be done while DB portion of patching is done. Both OAD_TOP and APPL_TOP should be copied from staged applications tier to production applications tier. However, the following configuration files should not be copied from staged environment to actual production environment.

```

$COMMON_TOP/util/apache
$COMMON_TOP/admin/scripts
$APPL_TOP/admin/<SID>
$FND_TOP/out
$FND_TOP/log
$COMMON_TOP/html/bin/appsweb.cfg
$COMMON_TOP/html/US/ICXINDEX.htm
$COMMON_TOP/html/_pages
$APPL_TOP/log/
$APPL_TOP/patches/
$COMMON_TOP/html/iby_debug.log
$COMMON_TOP/html/iby_error.log

```

Copying of the file system can be done by using any of cp, rsync or rdist commands.

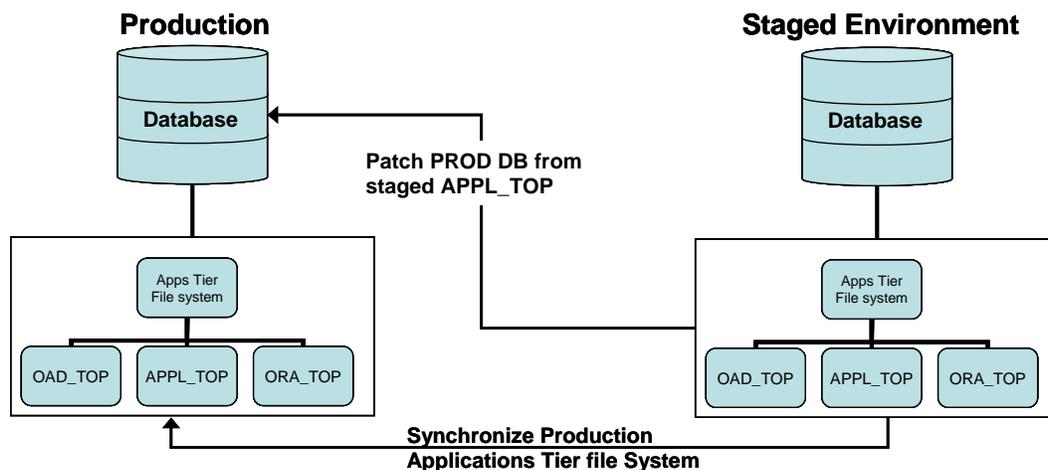


Figure 5: Applying patches to production environment and synchronize

Step 5 - Post tasks on production environment

Once DB portion of patches has been applied and applications tier has been synchronized, the following tasks should be performed on actual production system.

- Run autoconfig
- Import patch history

Usually, the bigger patches update the configuration templates and files at the end of patching. Production system gets updated configuration templates from staged environment. However, the configuration files need to be generated again by running autoconfig. Autoconfig recreates important files like start/stop scripts, appsweb.cfg, xml file etc. This step should be performed during the downtime only.

As you are applying the DB driver alone using adpatch, the production database will have DB portion of the patch history alone. So, you need to get the patch history for copy and generate portion from staged environment. The adphmigr.pl utility can be used to achieve the same. This step can be performed when the application is up and running. The steps are outlined below.

1. Export the copy and generate patch history from staged environment by using the following command.

```
perl $AD_TOP/bin/adphmigr.pl userid=apps/apps \  
startdate='2003/10/10 00:00:00' enddate='2003/14/10 00:00:00' \  
appsystemname=stage appltopname=tafnw1 nodatabaseportion=Y
```

The values against startdate and enddate should be replaced with the actual start and end dates of patching on staged environment. This command generates patch history files under \$APPL_TOP/admin/\$TWO_TASK directory of staged APPL_TOP. Two files are generated for each of the patch applied to the system.

2. Copy the patch history files from staged environment to actual production. The files should be copied from \$APPL_TOP/admin/\$TWO_TASK directory of staged environment to corresponding directory on production environment.
3. Import the patch history to production database. This can be achieved by invoking adpatch and answering all the questions until adpatch prompts for patch directory location. You need to type 'abort' at this prompt to come out of the adpatch. Adpatch uploads all the patching information from the files present under \$APPL_TOP/admin/\$TWO_TASK directory when is invoked. All the uploaded patch history files are deleted from this directory by adpatch itself.

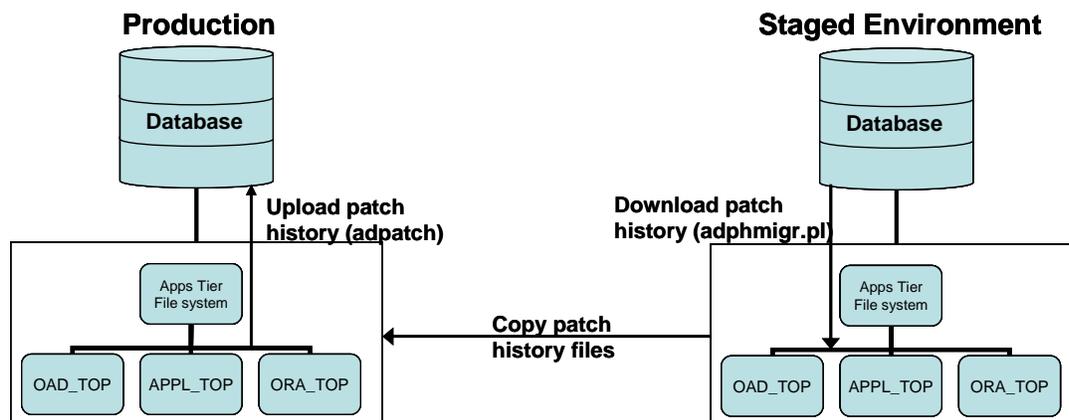


Figure 6: Post tasks on production environment

Best Practices

The staged APPL_TOP is specifically useful when multiple NLS languages are implemented or it is a major upgrade. In these cases, staged APPL_TOP approach should be considered a “must do”. You should try to implement the staged APPL_TOP methodology at the very early stage of your upgrade project and possibly from the first pilot upgrade run. It is need to test at least three or more cycles of upgrade to address all the issues and fine tune the timing of the upgrade project. There are some more tips which are helpful in Staged APPL_TOP approach.

- Staged environment should use different DB name and ports
- Test the complete patching multiple times if multiple family packs are applied
- Use this method only if copy and generate portion take significant amount of patching time
- Combine this approach with patch merge (admerge.pl)
- Gather database statistics with good sample size
- Use non interactive mode (defaultsfile=\$APPL_TOP/admin/\$TWO_TASK/adalldefaults.txt)
- Use shared APPL_TOP in case of multiple applications tier
- Check the time taken for copy and generate portion. This is the time saved by staged APPL_TOP approach effectively.

Limitations

You can not use the staged APPL_TOP approach in all kinds of upgrade projects. It is not useful if the downtime is not significant. There are also some other limitations, which are mentioned below.

- Staged APPL_TOP methodology can not be used for applying AD mini packs
- Staged APPL_TOP methodology can not be used for technology stack patches (ORA_TOP)
- If you are implementing the staged APPL_TOP methodology, you can not apply any patch to production environment directly after creating the staged environment
- Patches updating control file version of FNDLOAD should be merged together or applied outside the staged APPL_TOP

Conclusion

Staged APPL_TOP is an effective way to reduce the significant time during Oracle migrations go-live window. Many upgrades can be planned to go live on normal weekend, and organizations need not to wait for holiday weekend. It also provides opportunity to IT team as well as super users to validate the system in weekend itself before handing over to end users due to reduced downtime. The important aspect for the approach is to plan it at the early stages of the project, and performing multiple test cycles to make sure the effective implementation.

References

1. Metalink Note # 242480.1 - Using a Staged Applications 11i System to Reduce Patching Downtime
2. Metalink Note # 230672.1 - Cloning Oracle Applications Release 11i with Rapid Clone
3. Oracle Applications Maintenance Utilities (Part B10644-01 and B10641-01)