USING ENTERPRISE MANAGER TO ACHIEVE GRID AUTOMATION WITH *DEPLOYMENT PROCEDURES*

Deployment Procedures are Oracle's latest contribution in automating operations around the grid. This FAQ style technical document introduces the concept of Deployment Procedures to System Administrators and Integrators. The document spells out the advantages, the features and dives deep into few use cases that these Deployment Procedures are designed to solve.

Table of Contents

1. What are Deployment Procedures?

Deployment procedures are out of box best practices that comprise enumeration of a set of steps that are orchestrated by Enterprise Manager. Oracle ships a set of best practices Deployment Procedures to accomplish provisioning and patching related tasks. Deployment Procedures can be extended and customized for customer needs. The Deployment Procedure to patch a single instance database differs from the one to patch a RAC environment or an Application Server. Deployment Procedures can vary from one customer to another or a test installation to a production installation.

Deployment Procedures take into account and resolve the reality that environments are often different with each having complexities across different tiers with multiple dependencies. The situation is further compounded by existing operational practices. In a typical data center, Deployment Procedures can involve a design time activity (typically performed by a Lead Administrator) and a runtime activity (typically performed by the Operator).

Deployment Procedures have been introduced in Grid Control 10.2.0.2 and is licensed under the Enterprise Manager Provisioning Pack.

2. What are the few key advantages of Deployment Procedures?

a. Extensible

The objective of Deployment procedures is to have as many best practice methods out of box as possible. In an ideal case the customer should be able to run the deployment procedures as-is against a set of targets. Oracle shipped best practices Deployment Procedure cannot be modified. The customer can create a copy of the Oracle shipped Deployment Procedure and modify the same to insert or delete steps and error handling modes (see answers to Question 10).

b. Reusable

Deployment procedures are reusable. The steps of the deployment procedure can be based against directives that are stored in the Software Library. The deployment procedures can also be exported and imported across environments. This implies that the deployment procedures once developed for a test environment need not be recreated for production environment.

c. Hot-pluggable

The out-of-box deployment procedures are metadata driven. So new sets of procedures can be added to the Enterprise Manager environment without any additional outage.

3. What are the Out-of-box deployment procedures?

In version 10.2.0.4 of the Oracle Enterprise Manager, the "Out-of-box" deployment procedures include the following:

Patching procedures for:

- Single instance database.
- Real Application Clusters (RAC) – Rolling and All Nodes modes patching.
- Clusterware – Rolling upgrade patching.
- Automated Storage Management patching.
- Application Server patching.
- Host operating system patching procedures for Linux, Solaris, and Windows.

Provisioning Procedures for:

- RAC provisioning.
- Single Click Extend Cluster.
- Delete / De Scale Cluster.
- Application Server and Applications provisioning.

4. Are there any pre-requisites for using the Deployment Procedures?

For pre-requisites refer to Section 10.2 – 'Deployment Procedure Requirements' under "Enterprise Manager Advanced Configuration" from the library list at
http://download-west.oracle.com/docs/cd/B16240_01/doc/nav/portal_booklist.htm

Other pre-requisites before proceeding with the deployment procedures:

i. Configure the software library before starting to use the deployment procedures in 10.2.0.4.
   Refer to chapter 18 – Using Software Library in the above document.
ii. Run the 'Refresh from Metalink' job. (See Question 5 for details).
iii. Run the 'OPatch Update' job. (See Question 6 for details)

(The requisites are also mentioned in the Metalink Note: 427577.1, which also provides pointers to any additional info required)

5. What does 'RefreshFromMetalink' do? How to run it?

'RefreshFromMetalink' job downloads the metadata of the latest patch advisories, products and product versions available from Metalink. Additionally, it also does the computation of the vulnerable targets to the latest Critical Patch Updates available. (For example: Running the job post April 17th will generate advisories based on April CPU)

'RefreshFromMetalink' job can runs by default every 24hrs and updates from the Metalink. It is a requisite to run this job before proceeding to any patching or provisioning exercises. 'Refresh from Metalink' job can be run manually both in online and offline modes, its explained with illustration in the following section.

Online / Connected Mode:
The 'RefreshFromMetalink' job to can be kicked off manually by creating a job from the 'Job Activity' page under 'Jobs' tab.
The pre-requisite for the job to run is to pre-set up Metalink credentials. To setup, Click on 'Setup' link at the top and set the Metalink credentials in the 'Patching Setup' page.
The following figure shows, the job to be created, which would download the metadata from Metalink.

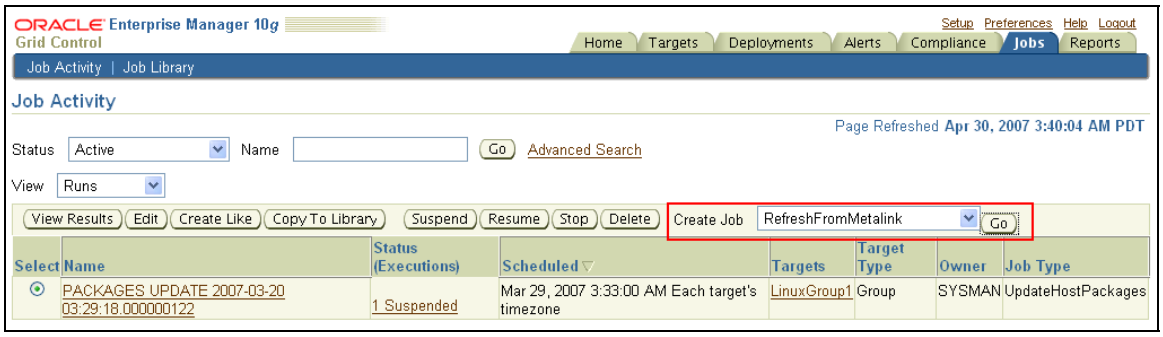


Figure 1: Manually Create 'RefreshFromMetalink' job.

Offline /Disconnected Mode:

The pre-requisite to run the 'RefreshFromMetalink' job in offline mode is to have the xml's required for the execution to be uploaded manually to the OMS. The following steps illustrates on how to execute the job in offline mode.

1) The following xml's are required to be downloaded from Metalink. Use a box, which has an Internet connection, download the xml's by providing Metalink credentials.

http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_products
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_platforms
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_languages
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_product_groups
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_product_releases
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_component_releases
http://updates.oracle.com/ARULink/XMLAPI/query_advisories
http://updates.oracle.com/ARULink/XMLAPI/download_seed_data?table=aru_releases

2) Upload the xml from the local box to the OMS from the 'Offline Patching Settings' page as shown in the figure below. Select the option 'No Connection' under the Metalink Connection Method.

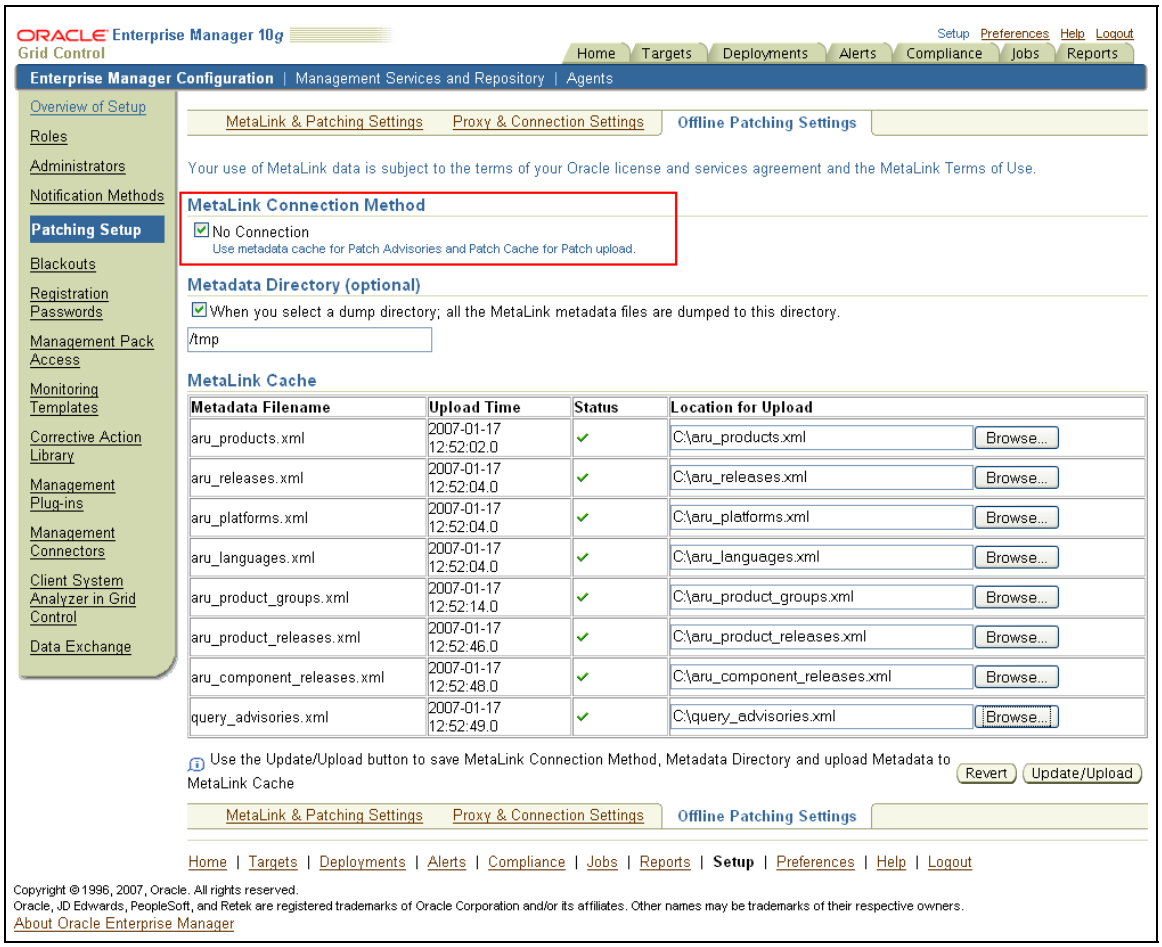


Figure 2:Offline Mode for Metadata Update

3) Create and execute the 'RefreshfromMetalink' job from the Jobs tab, as discussed in the online mode section above.

6. What does 'Upgrading Opatch' mean? What is its significance? How can one do 'OPatch Upgrade'?

Upgrading OPatch is to get the latest OPatch installed in the Oracle Home. Without the latest OPatch, certain bug fixes done in later OPatch releases will not be available to be used during the patching process. This can lead to patching failures in some cases. It is recommended to have the latest Opatch patches for successful operation of patching through the deployment procedures.

Opatch updates are released to Metalink through 2 different patches

- 4898608 – For versions 10.2.0.x and onwards.
  ARU Download URL: http://updates.oracle.com/download/4898608.html

- 2617419 –For versions lesser than 10.2.0.x.
  ARU Download URL http://updates.oracle.com/download/2617419.html

'Upgrade OPatch' is an automated step in the Deployment procedures, the step computes the appropriate version of Opatch picked from the software library against that of the version in the Oracle Home of the target, updates the Oracle Home OPatch, if it is of lower version.

The pre-requisite for the whole upgrade operation is to have the latest Opatch version patch in the Software library of the OMS. This can be achieved in two modes: Online and Offline mode.

The following illustration explains the two modes and the pre-requisites.

Online Mode:
The patches can be downloaded into the software library through the Opatch Upgrade job, which is to be executed once to download the Opatch patches from Metalink to the Software library.

The only pre-requisites for the job to run are to setup Metalink credentials. To setup, Click on 'Setup' link at the top and set the Metalink credentials in the 'Patching Setup' page.

The 'OPatch Upgrade' job to OPatch upgrade can be kicked off manually by creating a job from the 'Job Activity' page under 'Jobs' tab. The following figure shows, the job to be created, which would download the patches into the software library. This job is an automated job, which runs nightly, and will also be auto kicked off while setting up Software Library too.

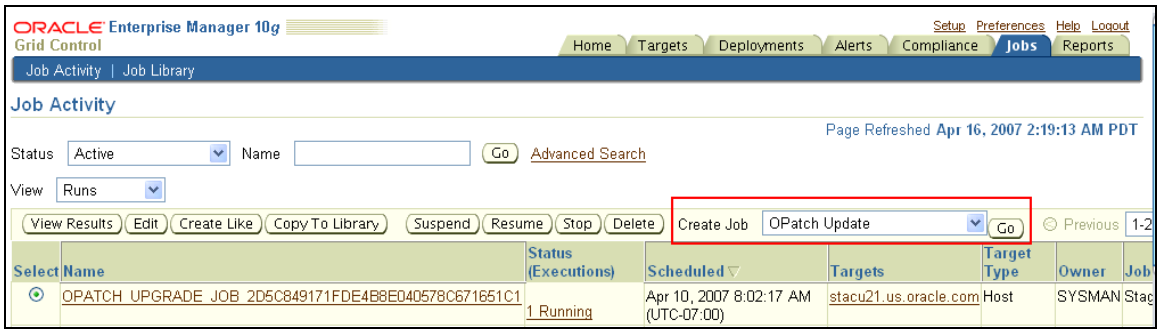


Figure 3:Create 'OPatch Update' job manually to download patches to software library.

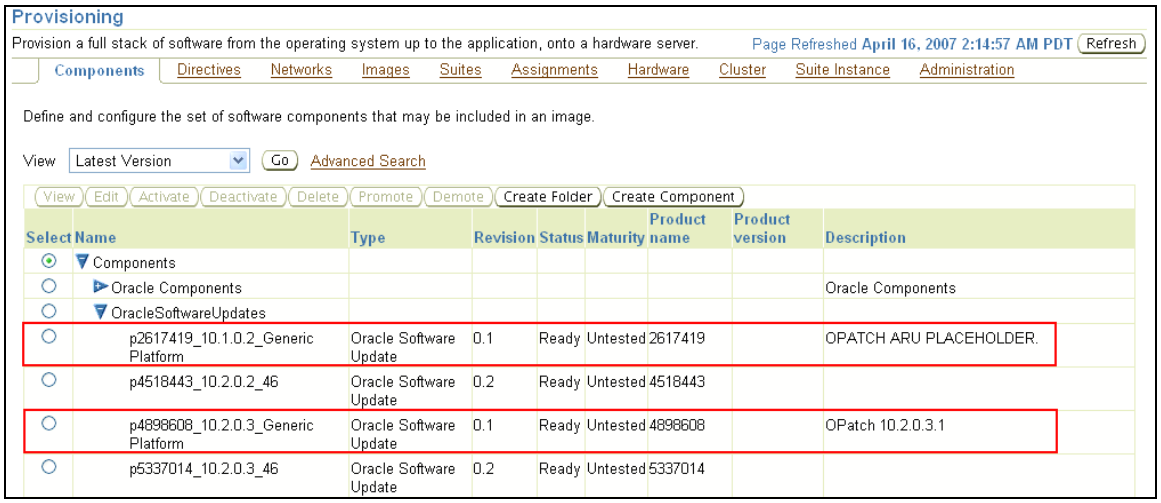


Figure 4:OPatch Place holder patches for version 10.2 onwards and below.

Offline Mode:

To get the patches to Software library in an Offline Mode, need to upload the patches to the software library and patch cache (optionally) manually from the local system.

From the Deployments page, click on link **View/Upload Patch** and click **Upload Patch**. Fill in the details required and upload the patch directly to the software library. The following figure illustrates it; provide the correct values to the parameters for uploading the patch. Make sure you select the Product Family – Oracle System Management Products and Product – Universal Installer.

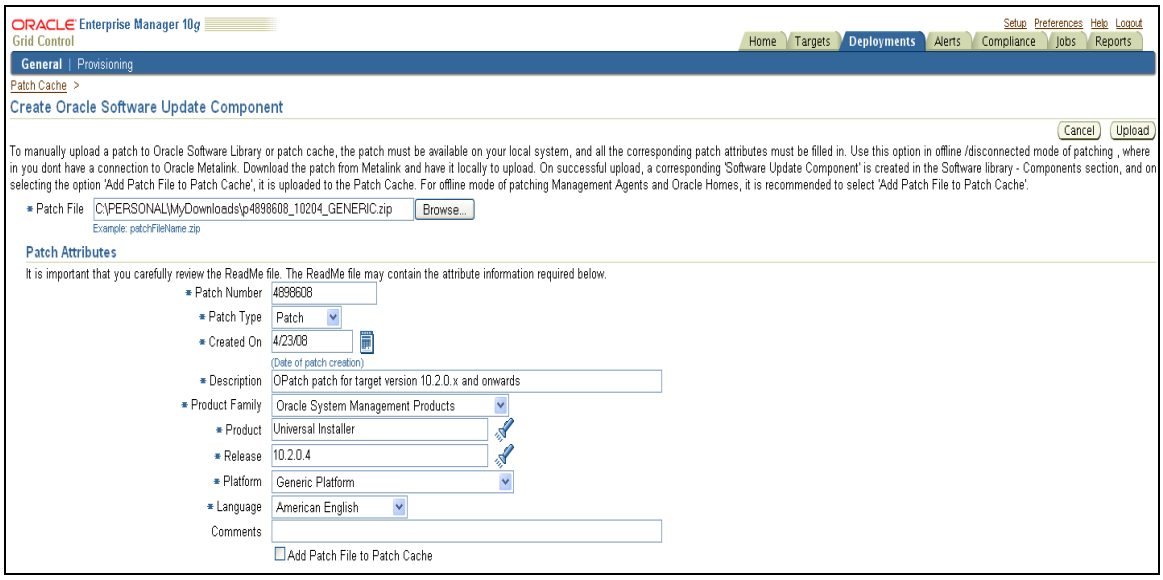


Figure 5:Upload the OPatch patches to the Software Library as a Software Update Component.

For version below 10.2, upload the 10.1.0.2 version of the patch -2617419 to the software library through the patch cache. Refer to the README of the patch for the information on the parameter values.

7. How can one use the out-of-box deployment procedure to apply a security related Critical Patch Update (CPU) to Oracle databases? Can I apply NApply CPU's too using the same procedure?

The out-of-box deployment procedure can be used 'as-it-is' to apply CPU to several single instance databases at the same time. If required the Out-Of-Box procedure can be customized as per the requirement, this involves a design time activity by a lead DBA that can be used over and over again by patch operators (Look for details of customizations under Question-10).

In 10.2.0.4 Oracle Enterprise Manager the standard procedures are capable of supporting the CPU's of type of Napply too.

In the following example patch 6914212–APR 2008 Critical Patch Update is applied to 10.2.0.3 databases.

This involves the following steps:

1) Select the 'Patch Oracle Database' Out-of-Box Procedure from the Deployment Manager page through the link **Patching through Deployment Procedures** under the Deployments Tab.

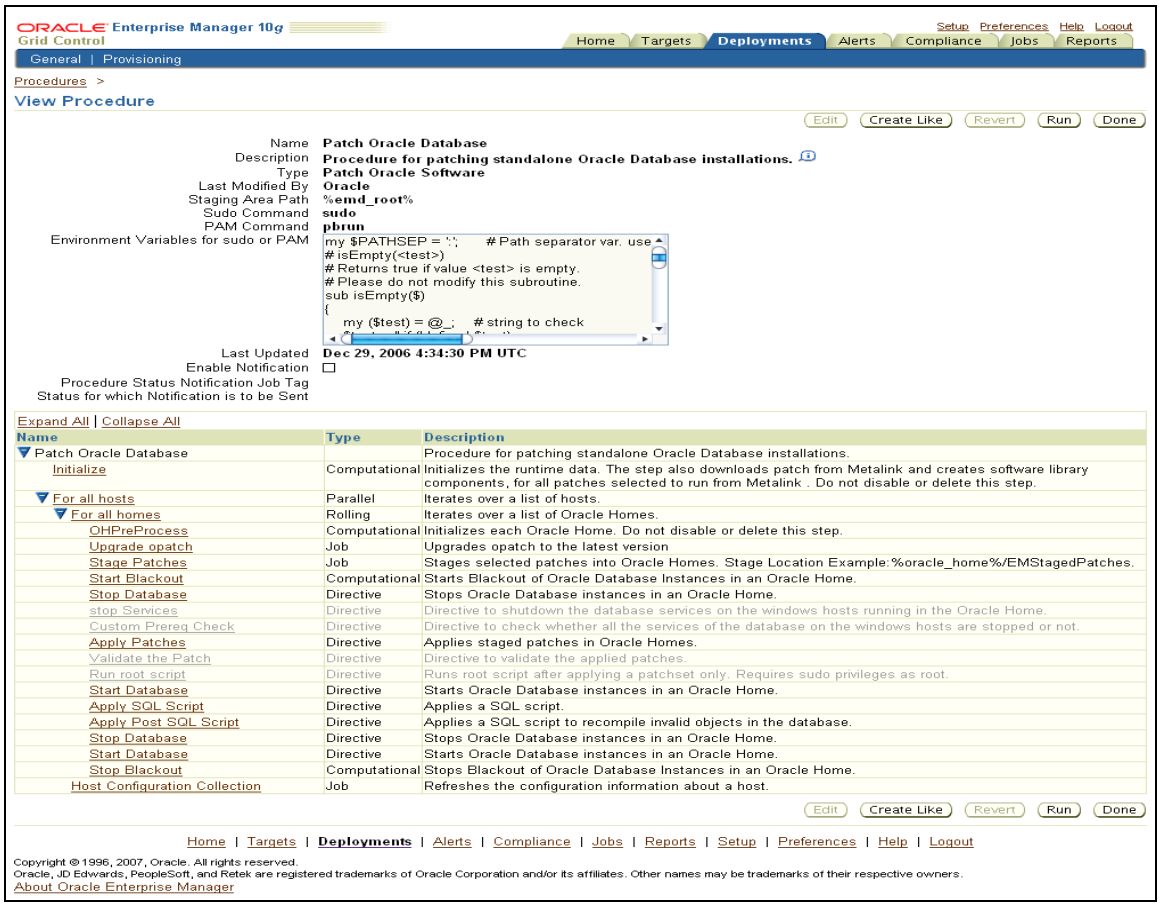


Figure 6:View the procedure and run it

2) Click **Schedule Deployment** and run it. There are 3 inputs that need to be provided at runtime-the patch number(s), the targets and the credentials. Finally the procedure needs to be scheduled for immediate or deferred execution.
3) One can choose to search the patch from Metalink or the Software Library. (Usage on 'Search from Software Library', check in Question 8). Here the patch 6914212is selected by Searching from Metalink.

Note: To search on Metalink, Metalink credentials have to be pre-set. To setup, Click **Setup** link at the top and set the Metalink credentials in the Patching Setup page.

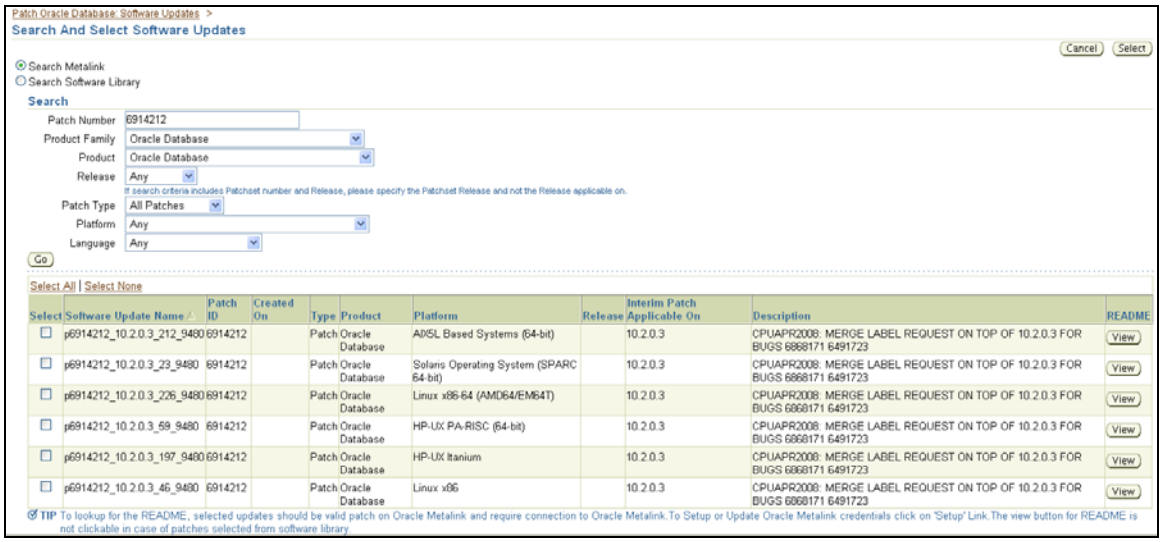


Figure 7:Search and Select the patch from Metalink

4) Deployment procedure by default runs the default SQL (catcpu.sql for CPUs) as shown below or you can optionally enter the location of the SQL you want run by choosing the other option “Enter the script to apply SQL”.

Additionally, ‘Upgrade OPatch’ can be made optional during run by unselecting the check box.

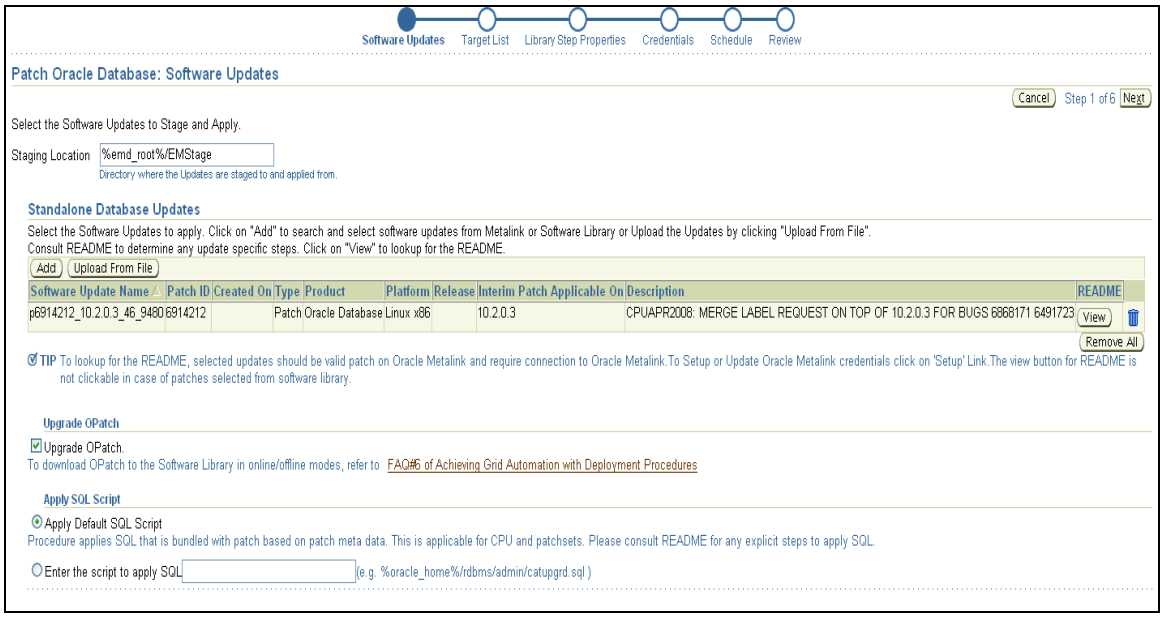


Figure 8:Choosing one or more patches from the Software Library or Metalink

5) The targets are then chosen. Search shows up one or more targets in the grid environment on which the patch is applicable.

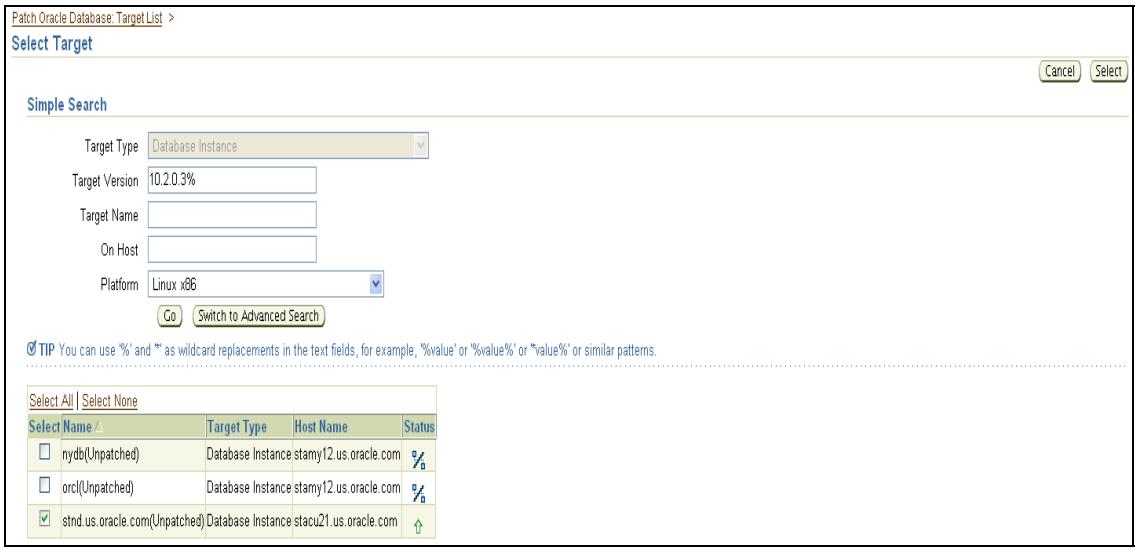


Figure 9:Choosing the list of Targets to patch

6) Followed by credentials. One can choose to use the ORACLE_HOME credentials from the repository by selecting the option **Use Oracle Home Preferred Credentials**.

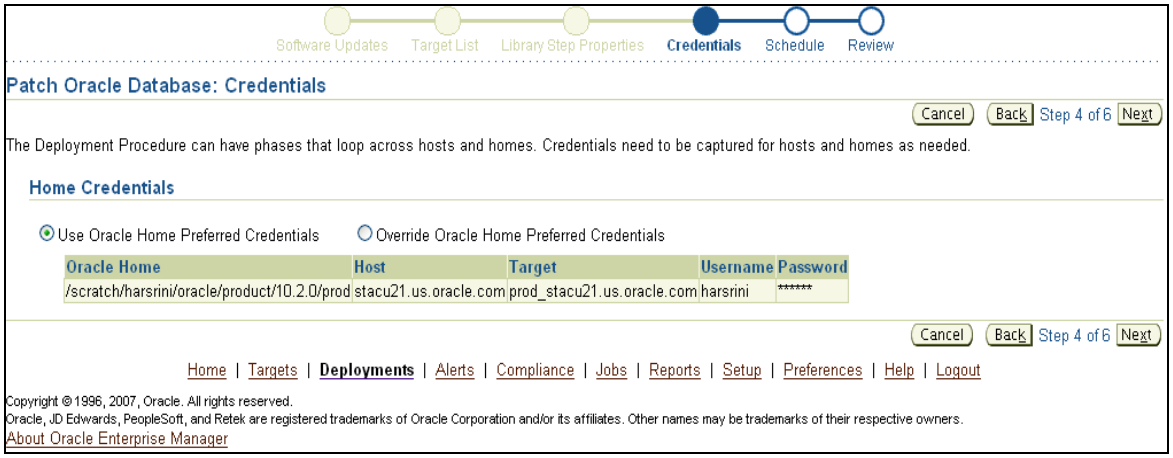


Figure 10:Specifying credentials

7) The procedure can be scheduled to run immediate or for later time.

On Submission, using **Procedure Completion Status** link can monitor the procedure run. It can be retried from a failed step, if required.

The flow for applying CPUs is illustrated in the following screenwatch:
http://download.oracle.com/enterprise_manager/10203_CPU_Patching/10203_CPU_Patching.html

8. What is Offline Patching? How can I do offline patching using Deployment Procedures?

"Offline Patching" is a mode of patching to address the case where you do not have a direct Internet connection to Metalink.
To perform Offline Patching through deployment procedures, you would require following the steps:

1. Download the Patch from Metalink from any box that has online connection and have it locally to upload to the Software Library.
2. Click on **View/Upload Patch** link from the Patching section under the Deployments Tab.
3. Select the **Upload Patch** option available in the Patch Cache.
4. Upload the patch zip file available locally and all mandatory attributes regarding the patch.
5. The patch is uploaded directly to the Software Library. Optionally in cases of Offline Agent Patching cases, the patch can be uploaded to Patch Cache by selecting the **Add Patch File to Patch Cache** check box. Refer to the patch README for the values of the parameters mentioned in the page.

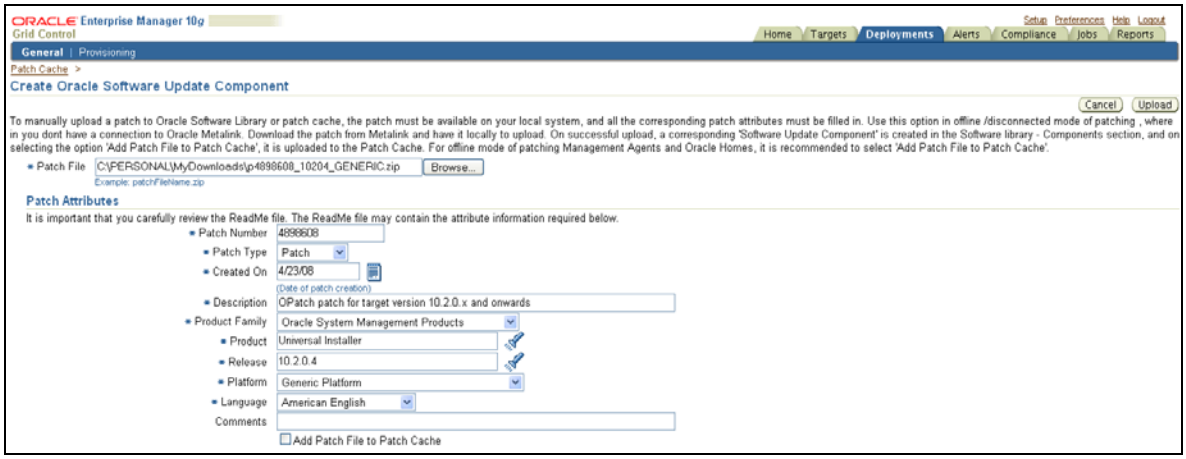


Figure 11:Upload patch to Software Library with all mandatory attributes.

6. Select the appropriate Deployment Procedure (Out –of –Box or Customized) based on the Targets to patch from the "Deployments Manager" page and run it.
7. While selecting the Software Updates, Choose to select from Software Library.

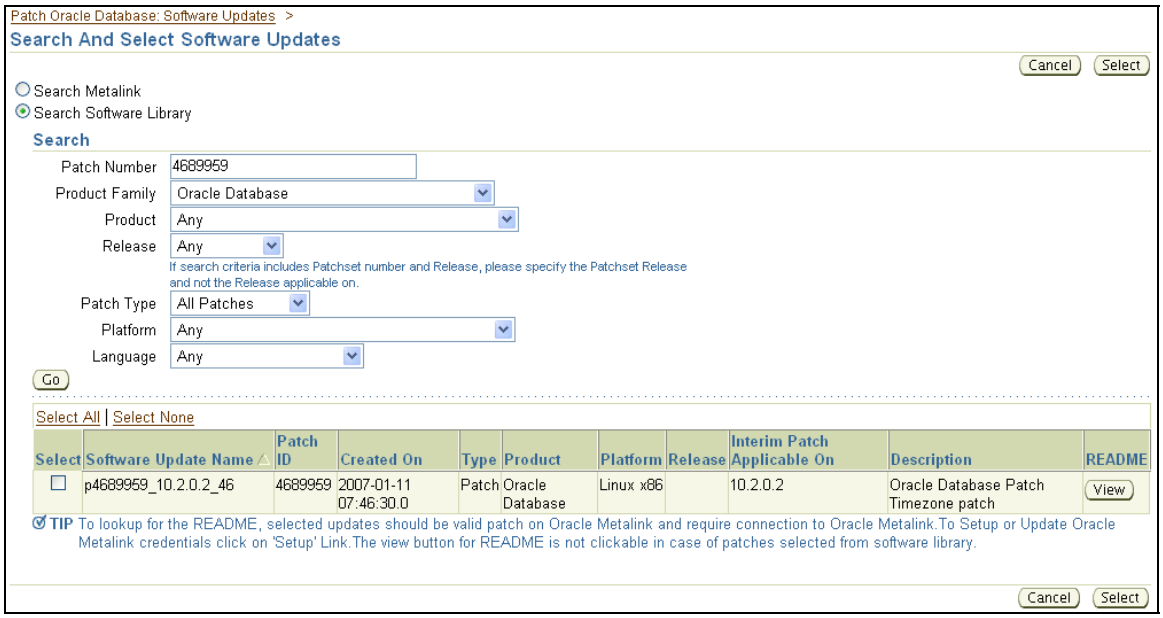


Figure 12:Select the option -'Search Software Library' for the Patch.

8. Proceed through the patching flow and patch the target. (Similar to the Question number: 7).

9. Can Deployment procedures be used to patch RAC in rolling mode?

Yes, deployment procedures support application of patches on Real Application Clusters in rolling mode.

Note: Read the README of the patch to check if it is applicable in rolling mode before proceeding. Critical Patch Updates are applicable in rolling mode.

The example below illustrates the application of an interim patch on the two nodes of a Real Application Cluster in a rolling mode.

1) From the Deployment Manager page select the out-of-box procedure Patch Oracle RAC Database –Rolling. The below figure, shows steps in the procedure for deletion of cluster. Click on **Run** to run it.

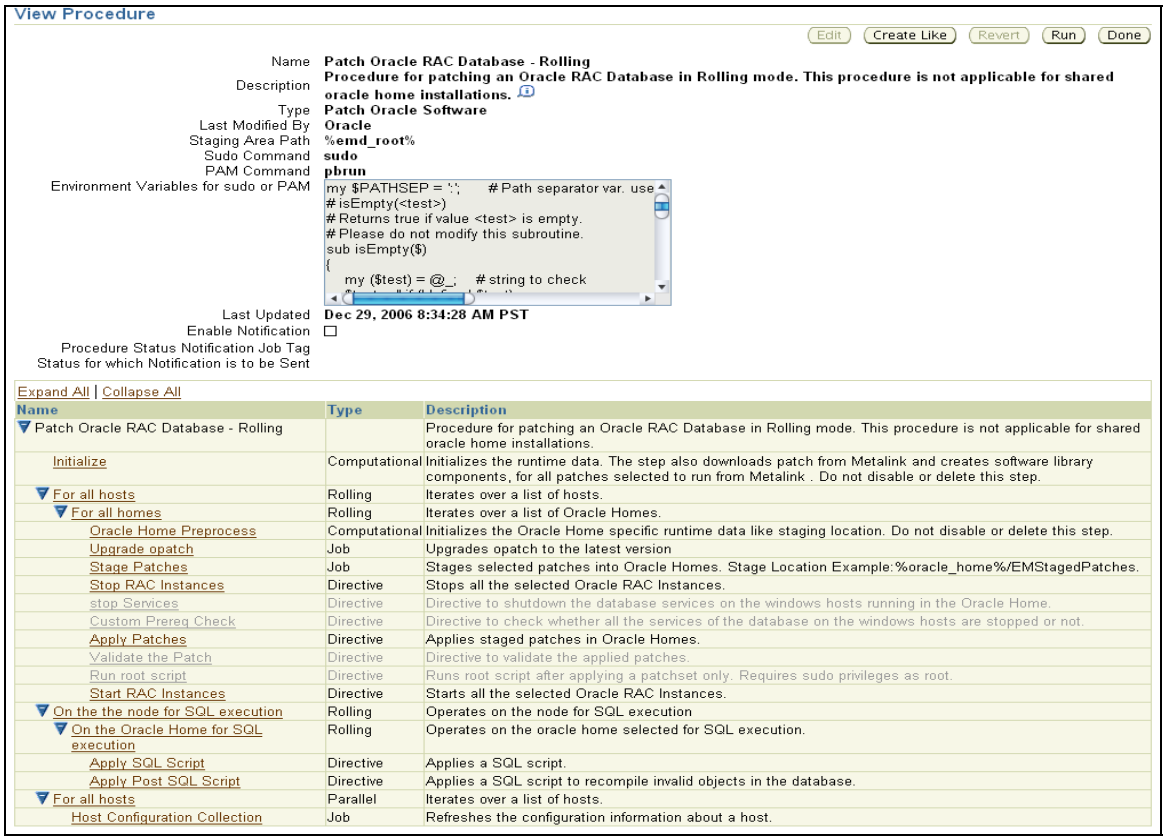


Figure 13: Patch Oracle RAC Database - Rolling Mode.

2) The first step is to select the Patches to be patched from either Metalink or Software library. In the example shown in the figure below, the patch number 4518443 is picked up form Metalink.

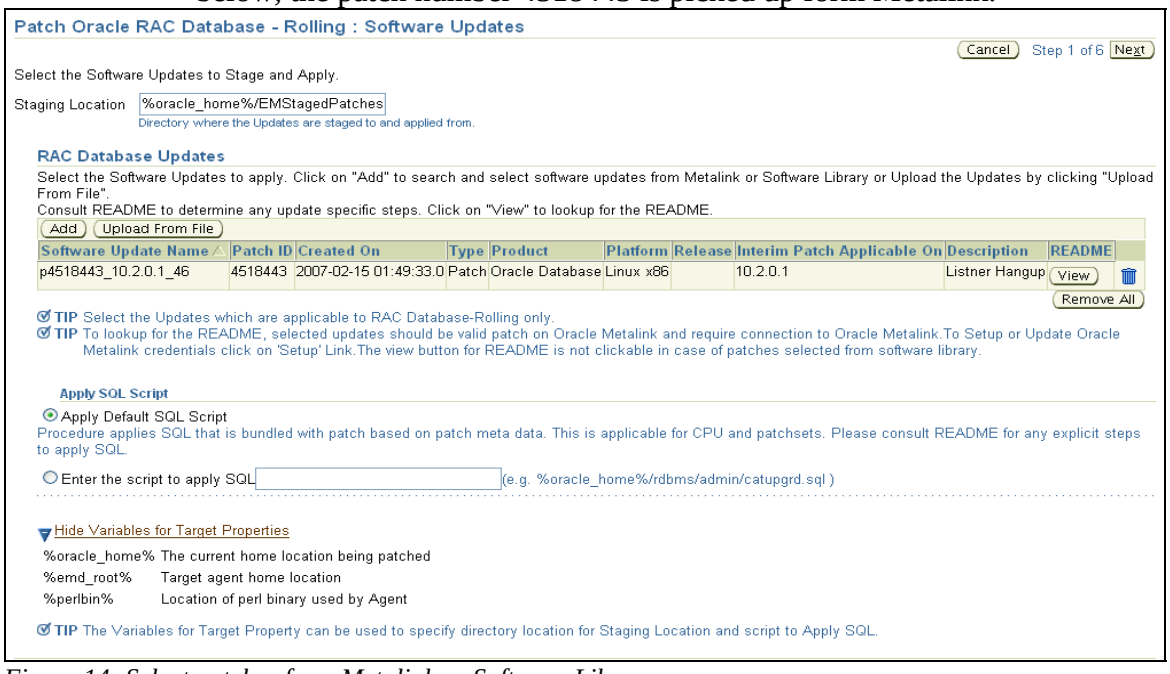


Figure 14: Select patches from Metalink or Software Library.

3) Proceed to the next step and select the Targets to be patched. The list of targets for which the patch is applicable is populated by default. On selecting the targets the next step shows up the instances associated with the target.

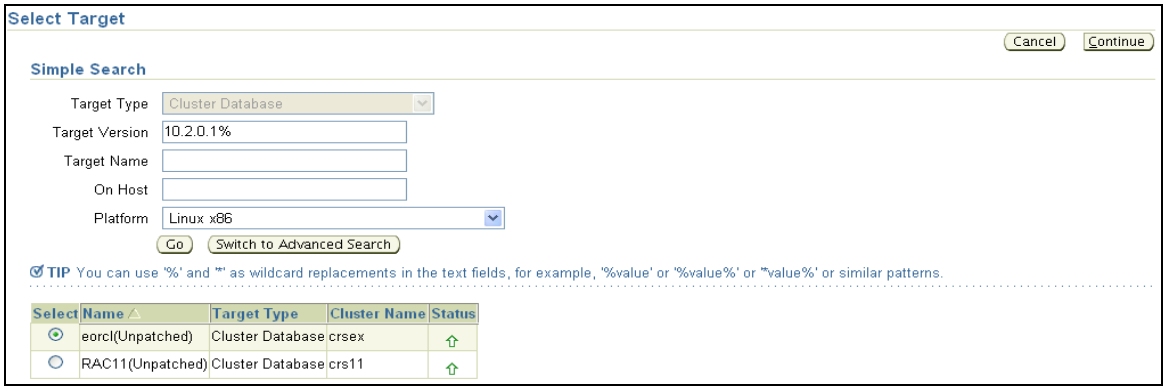


Figure 15: Select Targets from the Grid Control on which the patches are applicable.

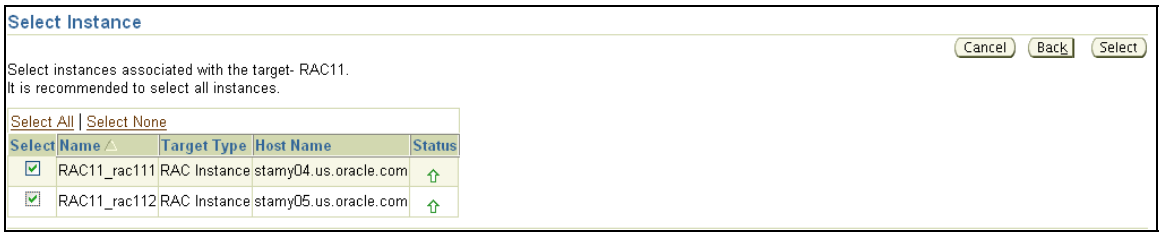


Figure 16: Select Instances associated with the Targets.

4) Proceed through and provide credentials for the targets and schedule the run. The status updates for the run can be checked through from the Procedure Completion Status page.

The figure below shows up the orchestration of the rolling patching on the nodes of the cluster database. In which the Apply patch step has shows up one node patched successfully and the application of the patch on the other node is in progress.

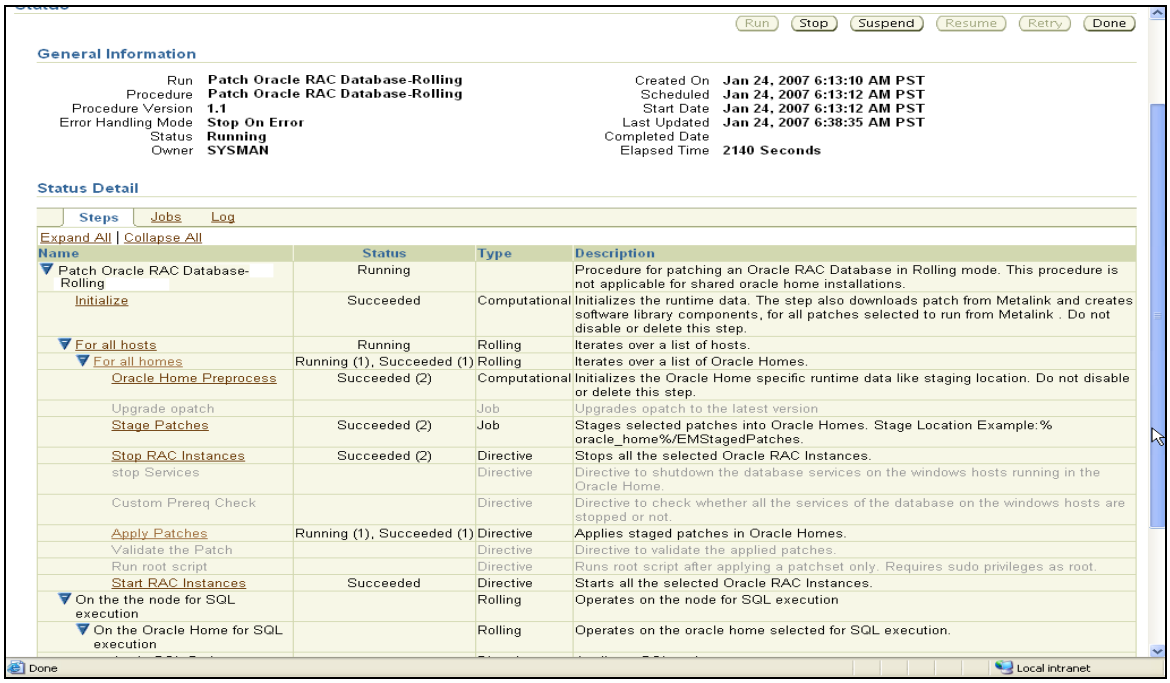


Figure 17: Patch Oracle RAC Database - Rolling mode.

Note: Almost all of the recent CPU's can be applied in a rolling fashion.
For applying Patchsets (For example 10.2.0.4 over 10.2.0.3) use the procedure 'Patch Oracle RAC Database – All Nodes'.

10. Are out of box Deployment Procedures customizable? If so, how?

Out-of-box Deployment Procedures are not customizable, but they can be used as starting templates to create similar procedures (using the Create Like functionality) that can then be customized. One can edit the deployment procedure to insert or delete a step or a phase or to enable or disable a step or a phase.

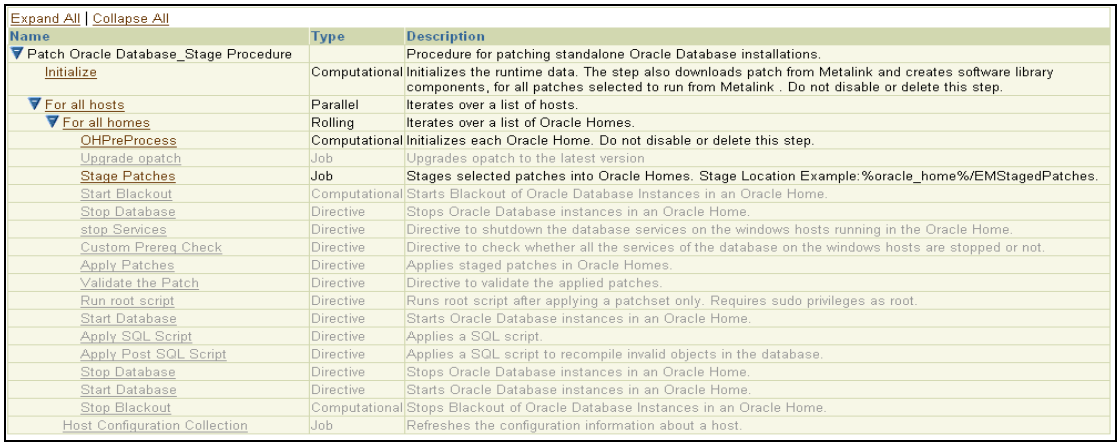
Deployment procedures also allow different error handling methods depending upon the case. For example, in a patching operation where hosts are patched in parallel, it may be wise to simply skip the host on which a failure occurs. However, failure on a device creation could render the remaining provisioning operation ineffective. So, it may be necessary to abort the entire procedure for failure of such a step.

Look for customization examples in the proceeding questions.

11. How can I use deployment procedures to handle patching in Low Bandwidth Data centers?

Deployment Procedure extensibility framework supports uploading the patch to a central location and patch multiple targets from a central location. This prevents the purpose of staging the patch on every target of execution, there by avoiding any network bandwidth related issues.
Follow the steps below to customize the deployment procedure.

1) Do 'Create like' of the out-of-box patching deployment procedures for example pick the 'Patch Oracle Database' procedure. Create a deployment procedure, which has only staging step in the deployment procedure by disabling all other steps as show in the figure below.

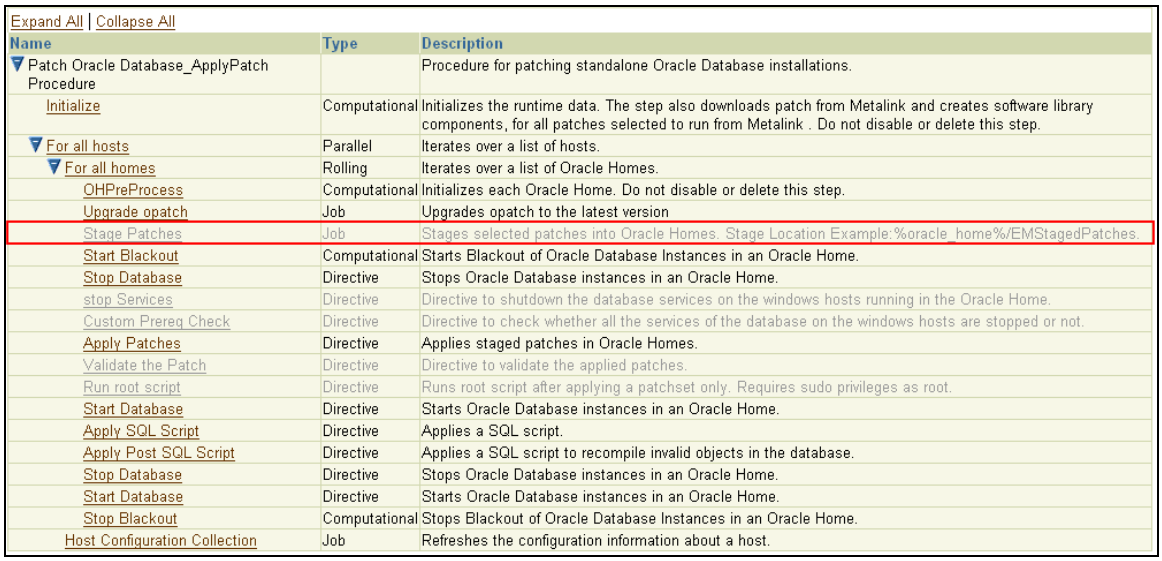


Expand All | Collapse All

Name	Type	Description
Patch Oracle Database_Stage Procedure		Procedure for patching standalone Oracle Database installations.
Initialize	Computational	Initializes the runtime data. The step also downloads patch from Metalink and creates software library components, for all patches selected to run from Metalink . Do not disable or delete this step.
For all hosts	Parallel	Iterates over a list of hosts.
For all homes	Rolling	Iterates over a list of Oracle Homes.
OHPreProcess	Computational	Initializes each Oracle Home. Do not disable or delete this step.
Upgrade opatch	Job	Upgrades opatch to the latest version
Stage Patches	Job	Stages selected patches into Oracle Homes. Stage Location Example:%oracle_home%/EMStagedPatches.
Start Blackout	Computational	Starts Blackout of Oracle Database Instances in an Oracle Home.
Stop Database	Directive	Stops Oracle Database instances in an Oracle Home.
stop Services	Directive	Directive to shutdown the database services on the windows hosts running in the Oracle Home.
Custom Prereq Check	Directive	Directive to check whether all the services of the database on the windows hosts are stopped or not.
Apply Patches	Directive	Applies staged patches in Oracle Homes.
Validate the Patch	Directive	Directive to validate the applied patches.
Run root script	Directive	Runs root script after applying a patchset only. Requires sudo privileges as root.
Start Database	Directive	Starts Oracle Database instances in an Oracle Home.
Apply SQL Script	Directive	Applies a SQL script.
Apply Post SQL Script	Directive	Applies a SQL script to recompile invalid objects in the database.
Stop Database	Directive	Stops Oracle Database instances in an Oracle Home.
Start Database	Directive	Starts Oracle Database instances in an Oracle Home.
Stop Blackout	Computational	Stops Blackout of Oracle Database Instances in an Oracle Home.
Host Configuration Collection	Job	Refreshes the configuration information about a host.

Figure 18: "Customized Stage Only" deployment procedure.

2) Repeat Step-1 and create a deployment procedure, which is an "Apply Patch" procedure, with the 'Stage Patches' step disabled. Figure below illustrates the same.



Expand All | Collapse All

Name	Type	Description
Patch Oracle Database_ApplyPatch Procedure		Procedure for patching standalone Oracle Database installations.
Initialize	Computational	Initializes the runtime data. The step also downloads patch from Metalink and creates software library components, for all patches selected to run from Metalink . Do not disable or delete this step.
For all hosts	Parallel	Iterates over a list of hosts.
For all homes	Rolling	Iterates over a list of Oracle Homes.
OHPreProcess	Computational	Initializes each Oracle Home. Do not disable or delete this step.
Upgrade opatch	Job	Upgrades opatch to the latest version
Stage Patches	Job	Stages selected patches into Oracle Homes. Stage Location Example:%oracle_home%/EMStagedPatches.
Start Blackout	Computational	Starts Blackout of Oracle Database Instances in an Oracle Home.
Stop Database	Directive	Stops Oracle Database instances in an Oracle Home.
stop Services	Directive	Directive to shutdown the database services on the windows hosts running in the Oracle Home.
Custom Prereq Check	Directive	Directive to check whether all the services of the database on the windows hosts are stopped or not.
Apply Patches	Directive	Applies staged patches in Oracle Homes.
Validate the Patch	Directive	Directive to validate the applied patches.
Run root script	Directive	Runs root script after applying a patchset only. Requires sudo privileges as root.
Start Database	Directive	Starts Oracle Database instances in an Oracle Home.
Apply SQL Script	Directive	Applies a SQL script.
Apply Post SQL Script	Directive	Applies a SQL script to recompile invalid objects in the database.
Stop Database	Directive	Stops Oracle Database instances in an Oracle Home.
Start Database	Directive	Starts Oracle Database instances in an Oracle Home.
Stop Blackout	Computational	Stops Blackout of Oracle Database Instances in an Oracle Home.
Host Configuration Collection	Job	Refreshes the configuration information about a host.

Figure 19:Customized "Apply Only" deployment procedure.

3) Run the saved customized stage only procedure of Step-1: "Patch Oracle Database_Stage_Procedure". In the first step of the run specify the central location where the patch is to be staged. The central location can be any NFS location or can be on a Net App filer, which is accessible by all targets to be patched. The below illustrates the place in the interview screen to specify the central staging location.

Figure 20:Specify a central location as the 'Staging Location'.

4) On successful execution of the step above, the patch will be staged in the central location specified. Proceed on to run the saved customized apply patch procedure of Step-2: "Patch Oracle Database_ApplyPatch Procedure". In the first step of the run specify the same staging location as provided in the above step.

5) Proceed on to select the patch, targets to patch and submit the procedure for execution.

The deployment procedure on execution patches the targets with the patch from the central location and not by the normal mode of pushing the patch to separate local staging location per target and patch from there.

12. What is a CRS Bundle patch? How can I apply patches of that kind to the Cluster Database through Deployment Procedures?

CRS Bundle patches are type of patches, which are applicable on both the CRS and RAC Homes on the nodes. For example patch number- 4091881, is a CRS bundle patch.
To apply the CRS bundle patches you would require to add customizations to the Clusterware patching deployment procedure shipped out-of-box (Patch Oracle Clusterware –Rolling), which would enable to apply the patch on the RAC instances running locally after patching Clusterware. A sample directive to apply patch on Multiple RAC instances is available in the Directives section of Software Library.

Note: Always refer through the patch ReadMe before using the process, if it has any additional steps incorporate that as a customized step into the procedure.

The steps below illustrate how to customize the directive and 'Patch Clusterware deployment procedure' to apply patches of this kind.

Pre-requisite for the procedure:
The sample directive 'Apply Patch on Multiple Homes' is only a read only script. Need to edit to be used in the deployment procedure.

1) Click on Deployments tab and click the **Provisioning** link in the Deployments page. Select the sample directive from the 'Directives' page under 'Directives>Oracle Directives>Patching >CRS>All>Generic". See figure 49 below for navigation and sample directive 'Apply Patch on Multiple Homes'.

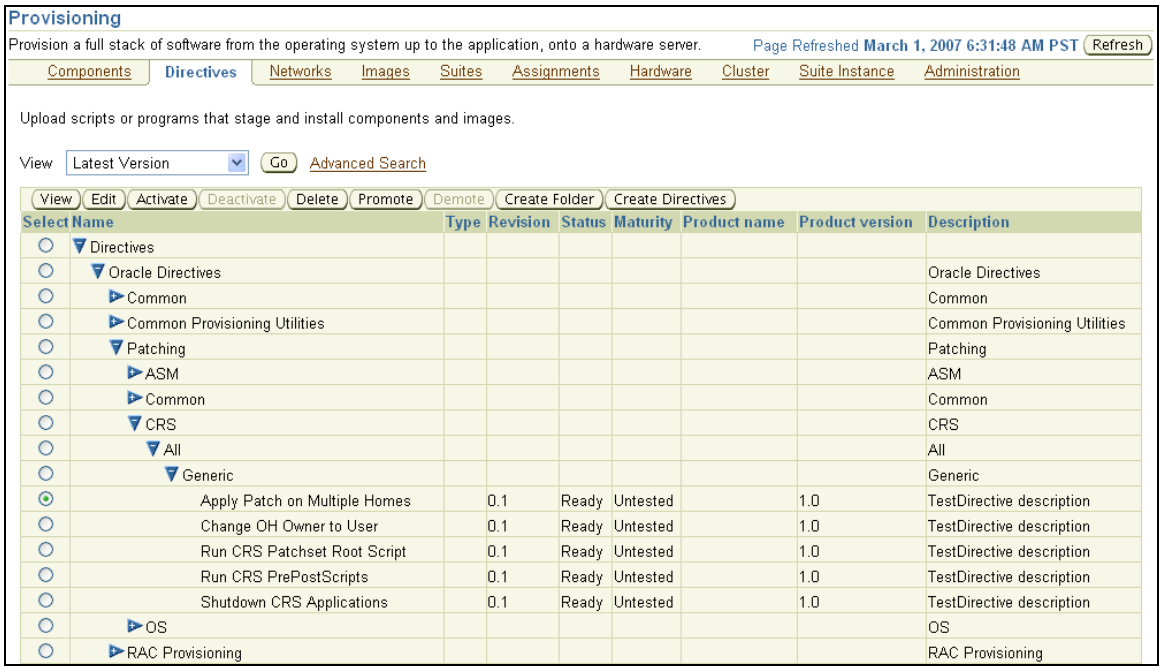


Figure 21: Sample directive 'Apply Patch on Multiple Homes.

2) Click on **Edit** to edit the directive, open the Perl file 'pa_patch_multiple_oraclehomes' under the Upload File section. Click Open the Perl file, and copy to a local editor (for example, Notepad). Edit the Perl file and uncomment the sections of the code which are under # Supporting Bundle Patch. See the figure below, which displays the Perl file of the sample directive to be edited.

Figure 22: Perl file associated with the Sample directive

Select the option **Upload from Local Machine** and upload the edited Perl file from the local machine. See the figure below, which shows up the option to upload the Perl file to the directive.

*Figure 23:*Upload edited Perl file to the directive from the Local Machine

Customizing the 'Patch Oracle Clusterware – Rolling' deployment procedure:

3) Do a 'Create like' of the out-of-box 'Patch Oracle Clusterware-Rolling' deployment procedure. Select the step **Apply CRS Patch** (See the figure below) and Click **Insert** to insert the custom step to apply patches of RAC Homes.

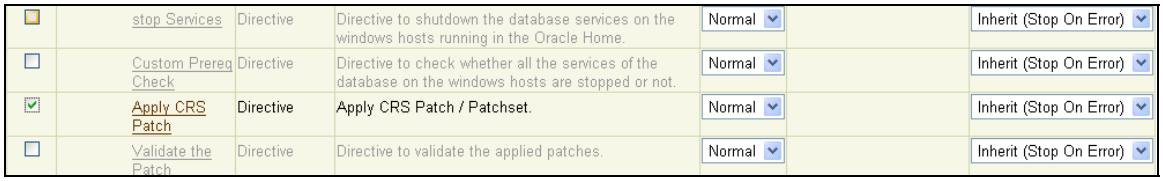


Figure 24: Select and Insert step below the 'Apply CRS Patch' step

4) Provide the required details in the Create section of the step. You must insert the Apply RAC patch after the Apply CRS Patch step. (See the figure below)

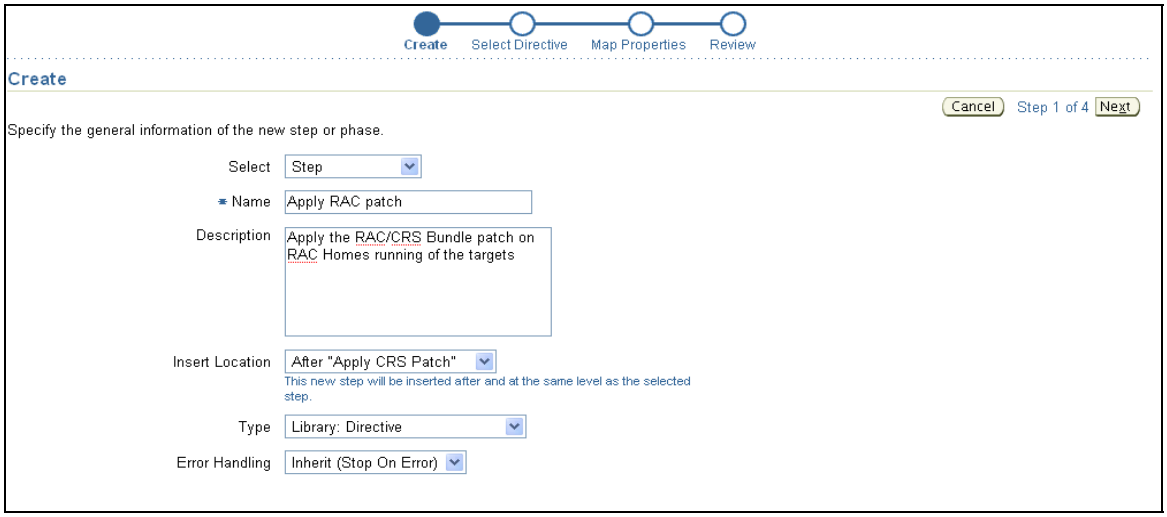


Figure 25: Create a custom step to apply patch on RAC

5) Select the Directive, which is uploaded to the Software library. See the figure below, which shows the selection of the "Apply Patch on Multiple

Homes" directive. Pick up the version 0.2 of the directive, which is the version of the edited copy of the standard sample directive.

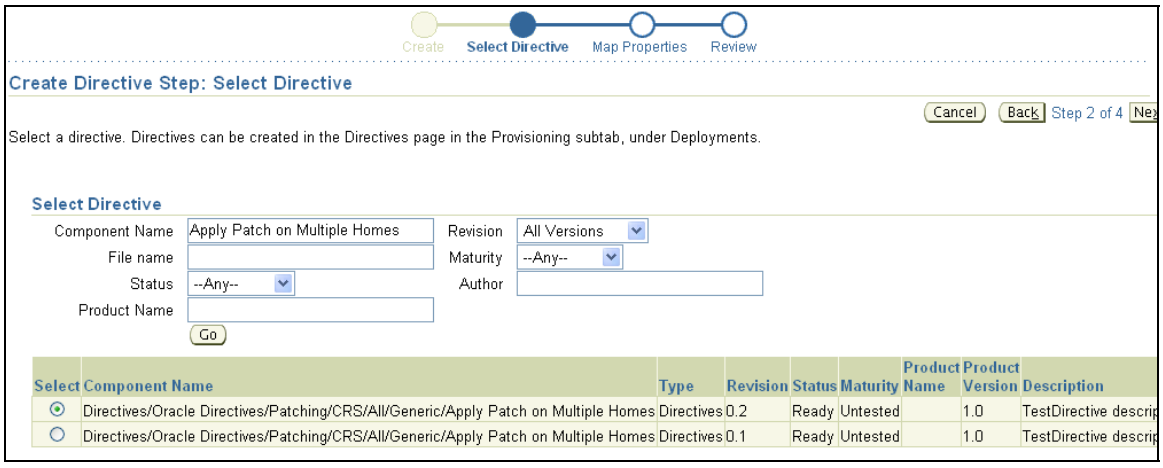


Figure 26: Select the new custom directive from the Software Library.

6) Proceed to the next step and Map the properties to the directive. See the figure below, which shows up the various mappings to the directive properties. Associate the RAC ORACL_HOME's to be patched to the exposed Target Variable – 'racLocalInstanceHomes'.

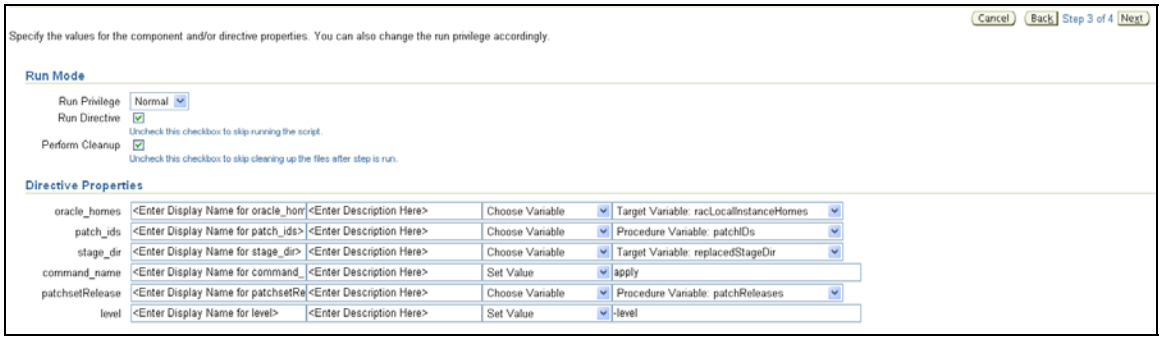


Figure 27: Map Properties for the 'Apply Patch on Multiple Homes' directive.

7) Proceed to review the selections for the new custom step to be incorporated. The figure below illustrates the review screen for the new Directive Step created.

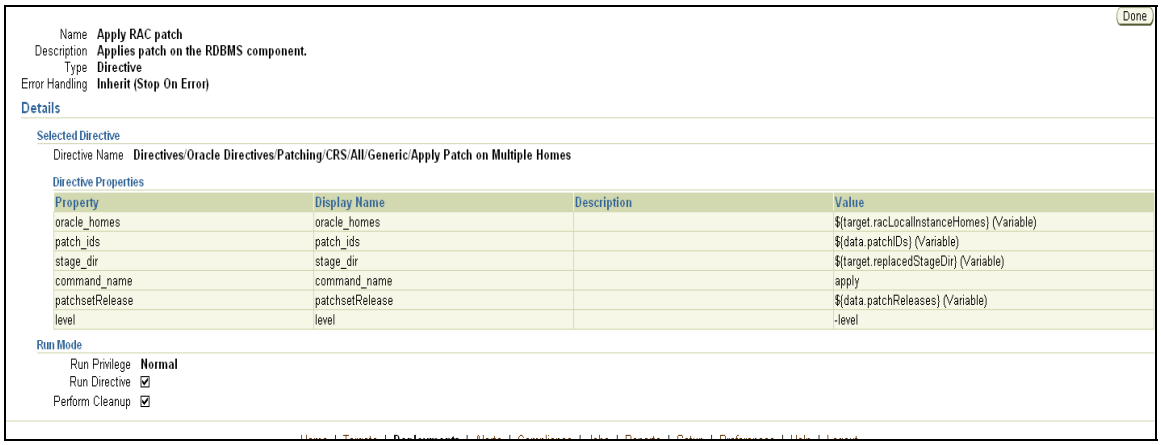


Figure 28: Review the created 'Directive Step' and add it .

8) Click **Finish** to incorporate the directive step into the deployment procedure. The figure below illustrates the step added to the deployment procedure.

Figure 29: Added directive step "Apply RAC patch" after "Apply CRS Patch" step.

9) Save the deployment procedure and 'Run' it. Select the CRS bundle patch to be used to patch in the Software Updates section. Proceed through the interview screens and provide in the required inputs for execution of the procedure.

Consult the patch 'readme.txt' of the patches to understand what steps need to be associated. In the above example as per the 'readme.txt' of the patch the only step, which required to be incorporated was to patch the RAC Homes present in the node. Incase of any extra steps; add those as directive steps to the deployment procedure in the same way as shown in the example above.

13. How can I add a node to my existing RAC cluster using the RAC deployment procedures?

You can add a node to the existing RAC cluster using RAC deployment procedures. Watch the screen watch explaining the usage of the deployment procedure to do this from the link below.
http://download.oracle.com/enterprise_manager/extend_cluster/extend_cluster.html

14. How can I delete my RAC cluster nodes using RAC deployment procedures?

You can select to delete or de-scale few nodes of the Cluster or the entire cluster using the out-of-box deployment procedures Delete / Scale down Oracle Real Application Clusters in a single click step.

Note: Delete /Scale down procedure supports delete node (of entire stack) only when there is only one RAC instance left. In multi instance case, use the Enterprise Manager delete instance functionality to delete one instance after other.

In the example below 1 node of 2 nodes cluster is deleted using the deployment procedure.

1) From the Deployment Manager page select the out-of-box procedure Delete /Scale down Oracle Real Application Clusters. The below figure, illustrates the steps in the procedure for deletion of cluster. Click **Run** to run it.

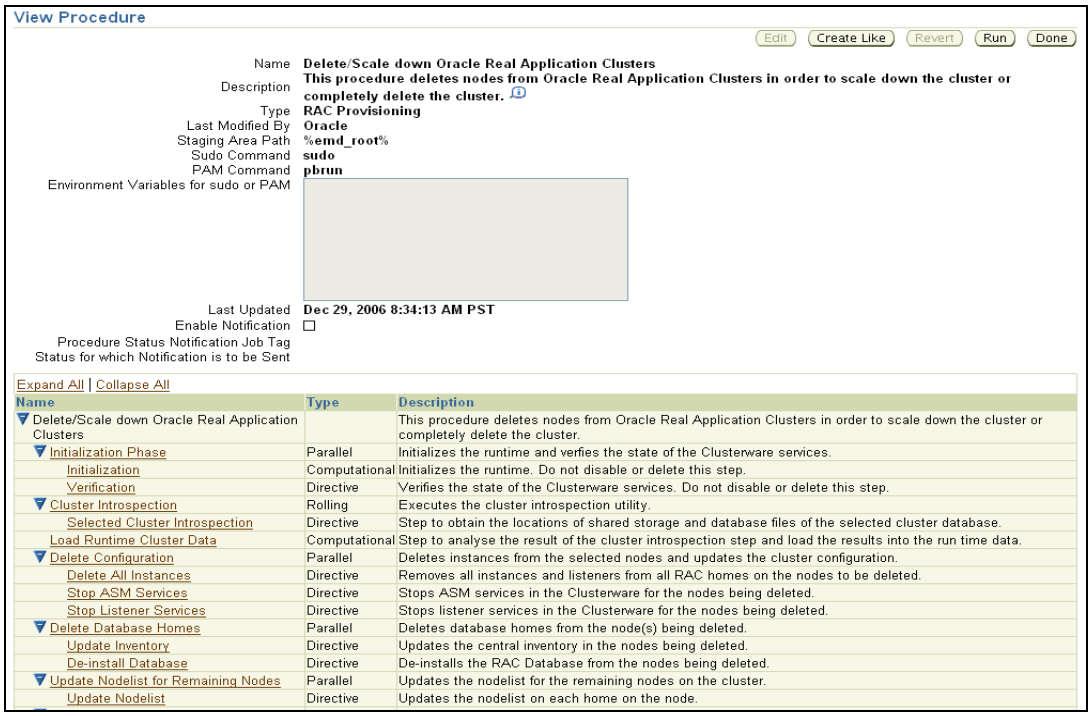


Figure 30: Delete/Scale Down Oracle Real Application Clusters deployment procedure.

2) The whole operation is on a single click .The page shows list of clusters available in the environment are showed up in the “Select RAC Cluster”. From the figure below, the existing cluster RAC 11 is selected. On Selecting the Cluster, the nodes corresponding to the cluster is populated in the ‘Available Nodes’ list. In this example one between the two nodes of the cluster is selected for deletion.

3) Provide the credentials for the cluster. By default, the operation is scheduled for immediate execution, edit if required, to change the schedule for a later date and time.

4) Click **Review** to review the selection and submit the procedure for execution. Check the progress of the execution form the Procedure Completion Status page.

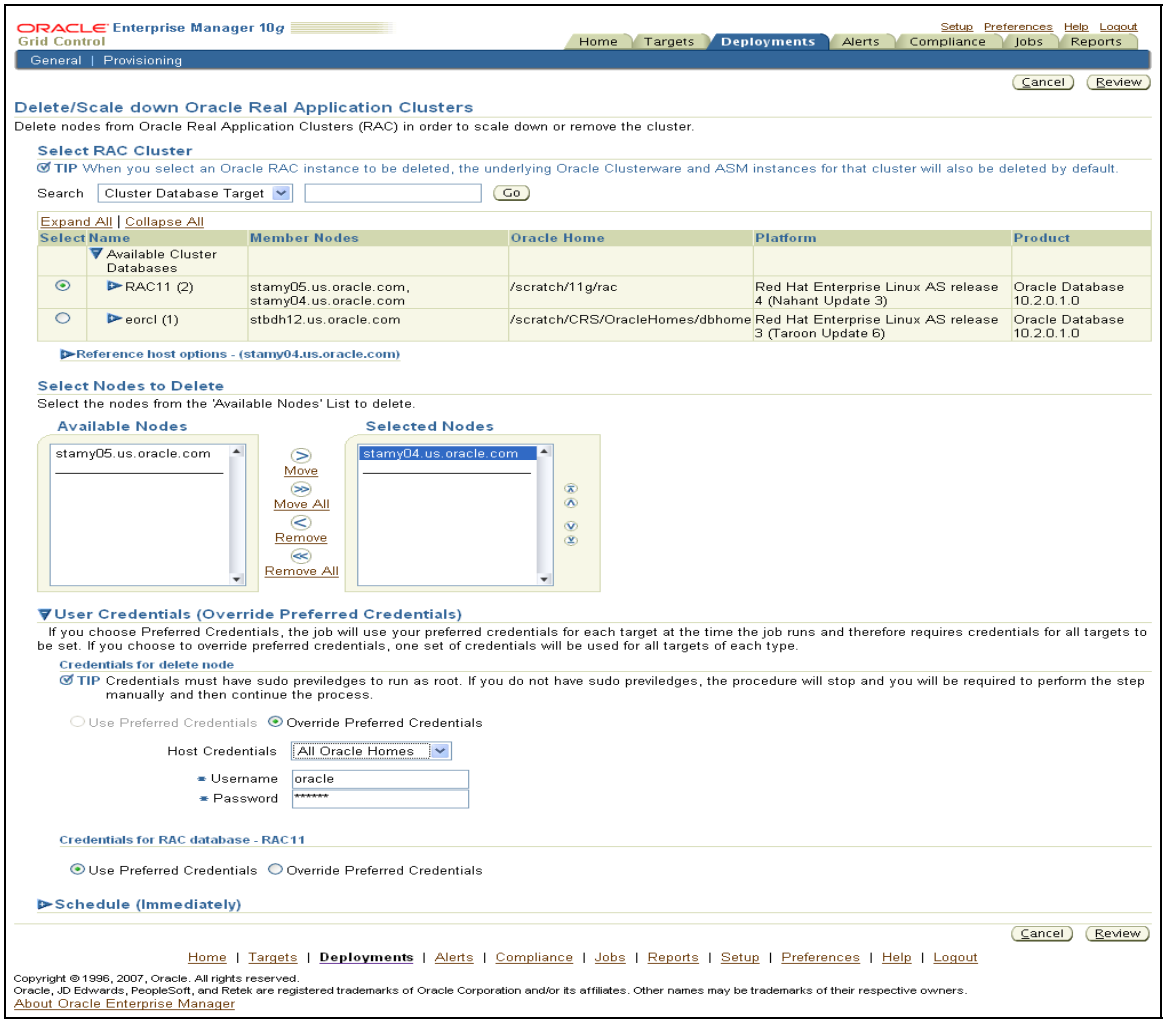


Figure 31: Select the nodes for deletion from the existing clusters.

15. How can I use deployment procedures to deploy a J2EE application on the application server node?

You can use deployment procedures to deploy you custom j2ee application on to a new cluster node by extending the existing AS cluster. The out-of–box deployment procedure for provisioning Application Server has steps, which allow deployment of custom j2ee applications. It

is disabled by default, on enabling and configuring you can deploy the custom application on the new node, which you are provisioning.
The following example shows you how to do to customize the deployment procedure and incorporate the application to be deployed on the new extended application server.

1) Do a 'Create Like' of the Out-Of-Box "Application Server Provisioning Deployment Procedure 10.1.3" deployment procedure.

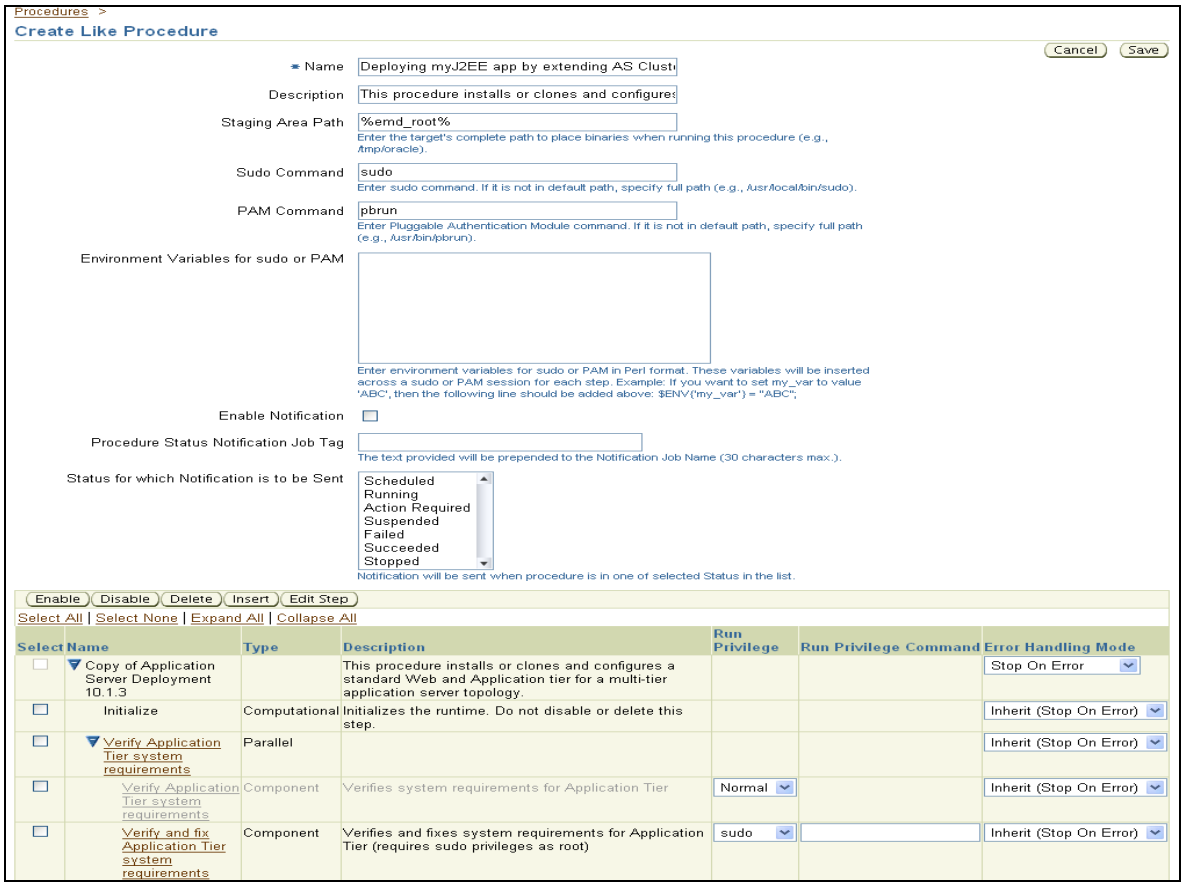


Figure 32: Copy of the Out-Of-Box Application Server Deployment 10.1.3 procedure

2) Select the step Deploy Application from the procedure and Enable it.

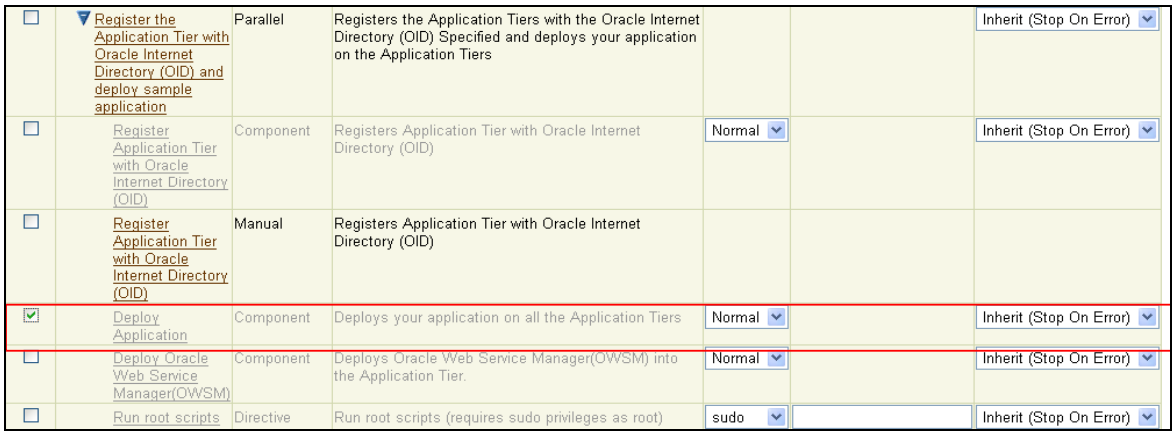


Figure 33: Enable 'Deploy Application' Step.

3) Select the step and click **Edit**, to incorporate the required application and directive to deploy.

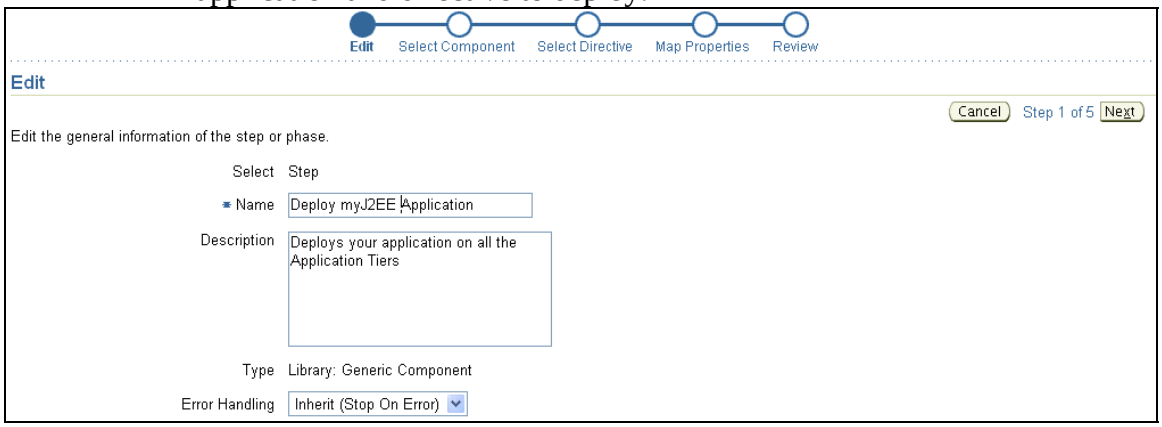


Figure 34: Edit the 'Deploy Application' step

1) Deploy application step is tied up with a sample application ear file. On proceeding to select Component step, you have options to **Retain Selection**, which is of the sample custom application or **Select the New Component**, which would provide options to pick components from the software library. The figure below illustrates you the options seen for the step.

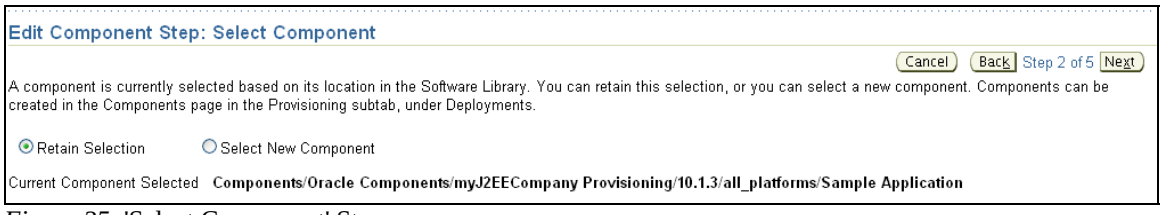


Figure 35: 'Select Component' Step

The custom j2ee application, which is to be deployed "myJ2EEApplication", is uploaded into the software library as a generic component.

2) Select “myJ2EEApplication” component uploaded to the software library.

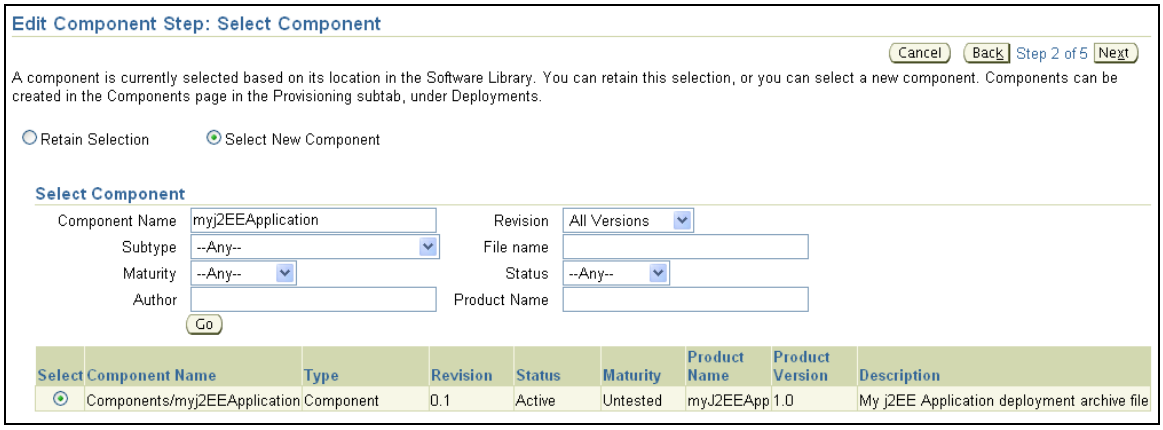


Figure 36: ‘Select New Component’ from Software Library

3) Proceed to select the directive to associate with the step for deployment of the custom application component. Directive to deploy J2EE applications comes out-of-box, but if you have custom script to carry out the deployment you can choose the same from the stored directives in the software library. For example, to deploy the custom component as shown in the figure below, selecting the option **Retain Selection** uses the directive shipped out-of-box.

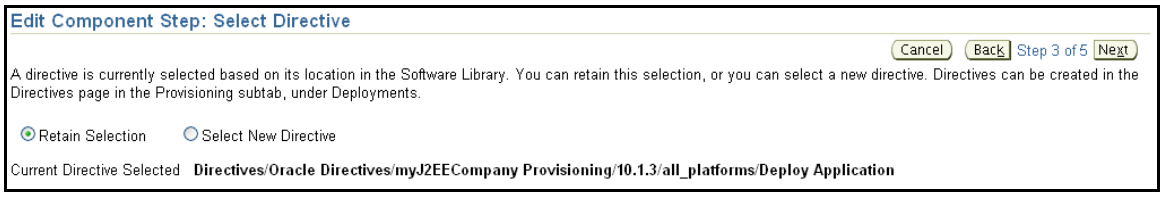


Figure 37: 'Select Directive - Retain Selection to use the out-of-box directive.

4) Proceed to next step to Map the properties required by the directive in deploying the component. In the example in figure below, the values for OC4J_ADMIN_USER, APPLICATION NAME, EAR FILE NAME, etc. is set and the ORACLE_HOME value will be picked up from the variable during runtime.

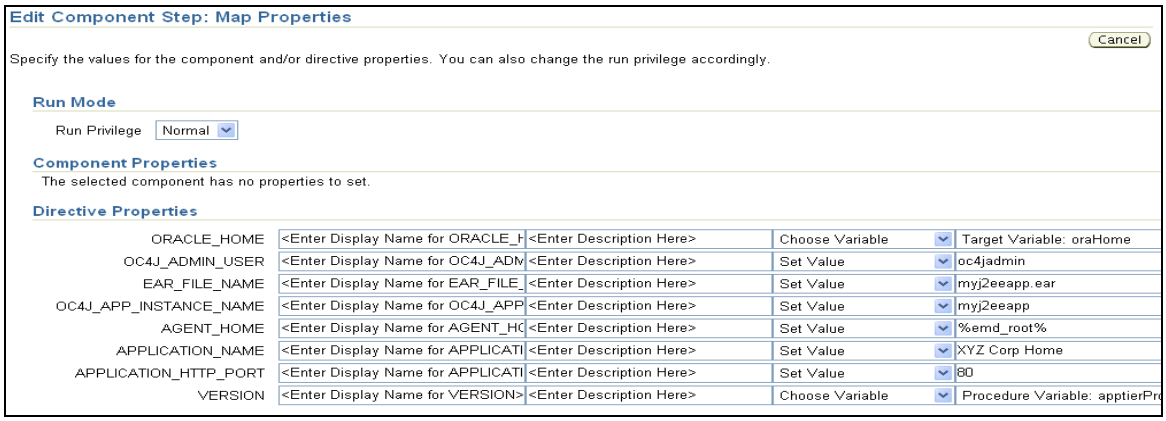


Figure 38: Map Properties for the directive

5) Proceed to Review step to review the selections and click **Finish** to add the Deploy Application step to the custom deployment procedure.

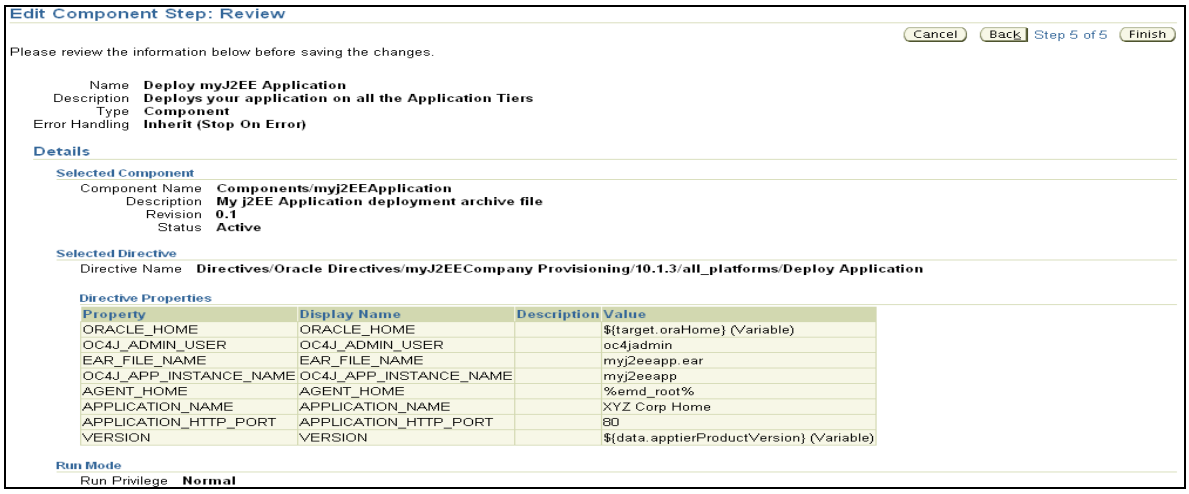


Figure 39: Review and Finish to add the step

6) 'Save' the custom deployment procedure and proceed to run it. On running the deployment procedure, it would take through review screens to provide in runtime values required for the run. The following steps illustrate how the 'Application Server Provisioning' deployment procedure will be executed.

7) First step in the process is to select the source. The source can be either a reference from an existing cluster or a certified, tested image of the application server in the software library. In the figure below, the source is selected from the existing installations shown up in the drop down list "Select Cluster".

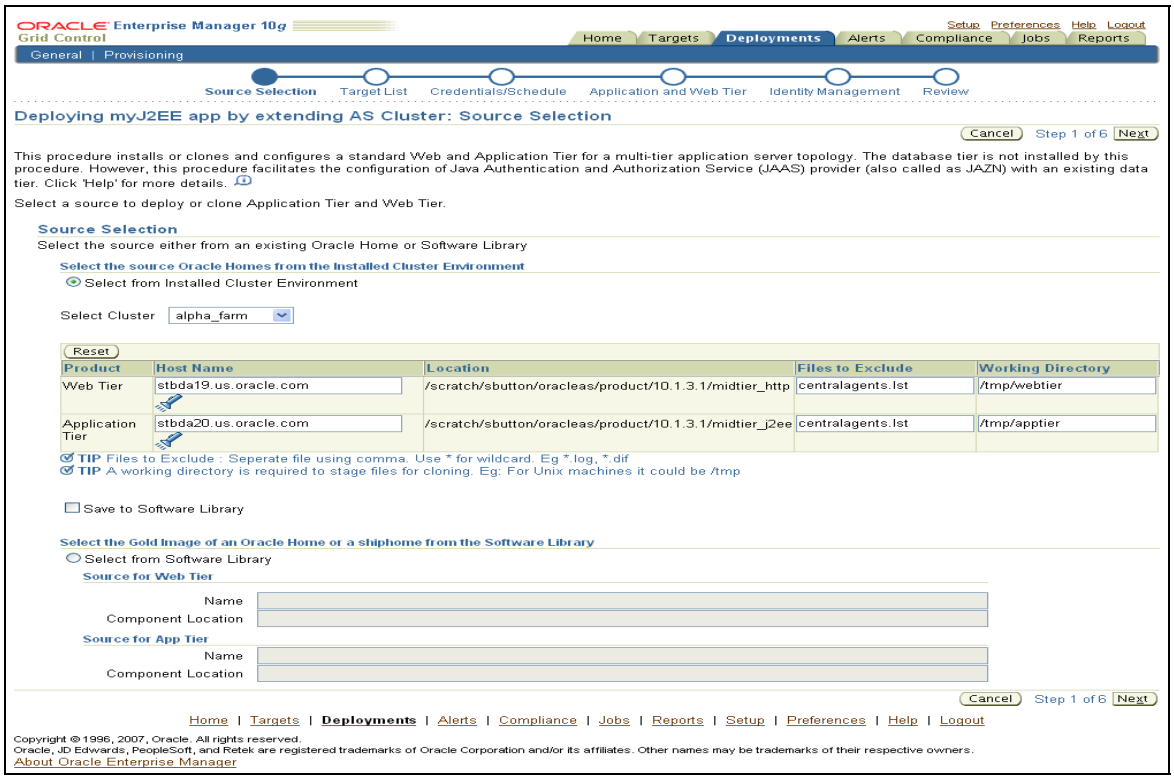


Figure 40: Select Source- Select from existing Cluster Environment.

8) Proceed to the next step to select the targets for Web Tier and Application Tiers. In the figure below, the destination targets are searched and selected from the Grid Control for Web Tier Hosts and Application Tier Hosts.

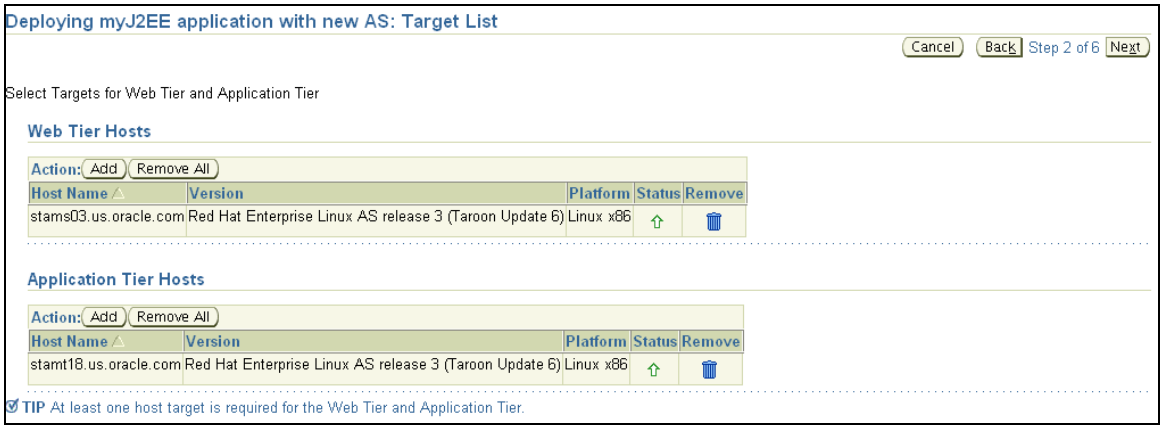


Figure 41: Select Targets for Web Tier and Application Tier

9) Provide the credentials for the Target and Source and Schedule the procedure in the next step. In this example, the required credentials are provided and scheduled for immediate deployment.

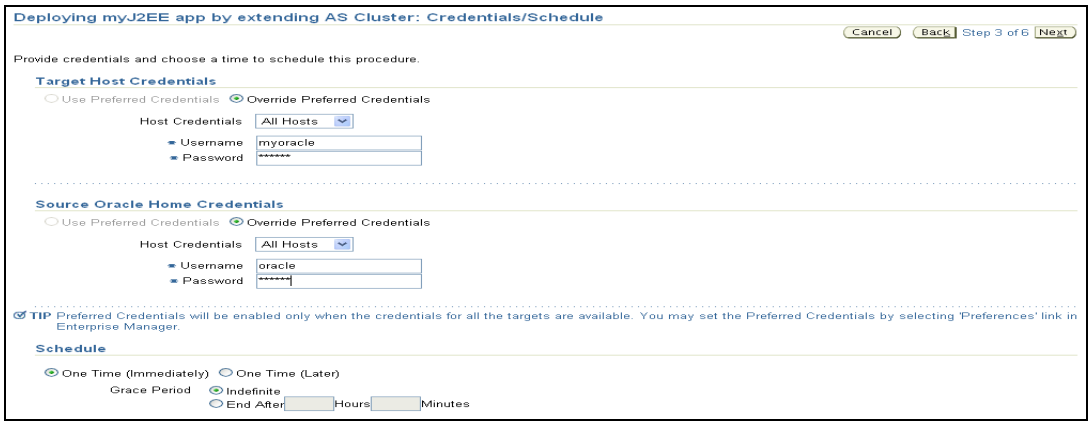


Figure 42: Credentials for Source and Target. Schedule the procedure for execution.

10) Proceed to the next step and provide in other configuration details required for the extension of the existing cluster. In this example, as in the figure below the configuration details like Instance details, Port details, and Load Balancer Details and ID management configuration details are provided.

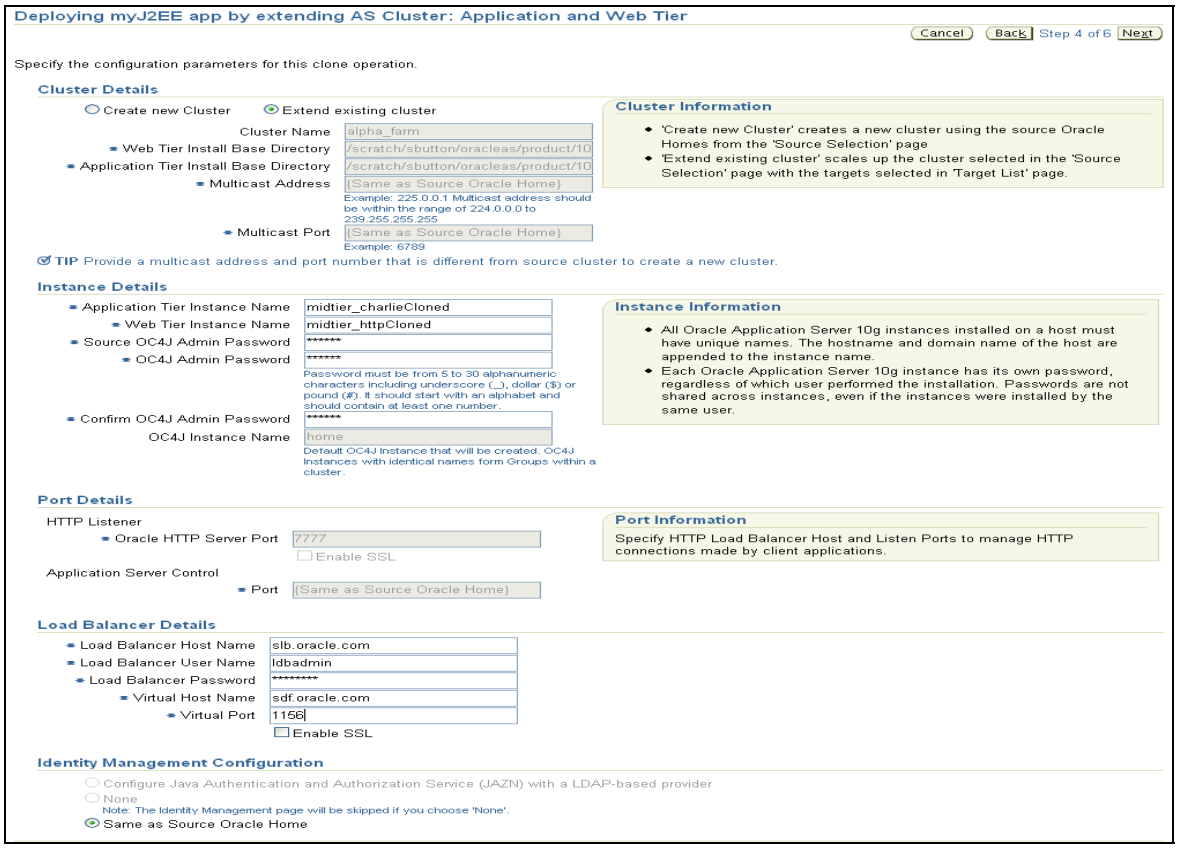


Figure 43: Configuration Details required for the extension of the existing cluster.

11) Proceed to review the selection. Submit the procedure and check the progress of the execution from the Procedure Completion Status page.

16. What are examples of customizations that a customer can incorporate in a patching deployment procedure?

Following are three examples how deployment procedures can be customized:

a. Manual step

XYZ Corporation has a process of making sure that users are logged off from their application before the database is shutdown. The DBA checks with key users that they have indeed logged off before proceeding with the Database shutdown. This can be achieved by introducing a manual step before the "Stop Database" step. The procedure would pause on the completion of the manual step. Only when the DBA chooses to continue the procedure would advance. The following screenshot shows how manual step can be inserted into a procedure.

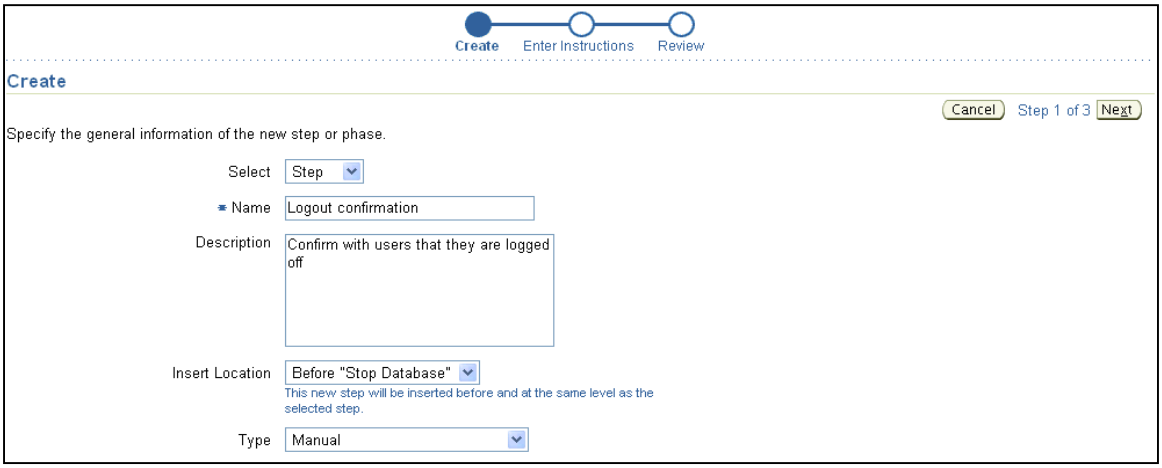


Figure 44: Inserting a manual step to confirm users are logged-off

b. Application service shutdown and startup handling

Deployment procedures can be used to perform operations that are outside the scope of out-of-box procedures. Examples include stopping and starting an ERP application or registering a newly provisioned service with the load balancer. Each of these steps can run in the context of any valid operating system user and can make use of Pluggable Authentication Module like "pbrun" (Powerbroker). They can also run in super user mode using "sudo" (see drop down for "Run Privilege" in figure below).

In the following example, the deployment procedure shuts down the Oracle Internet Directory (OID) service before patching the underlying database and starts it up (not shown) after the patching. The OID shutdown step has been placed in a phase for ORACLE_HOMEs

containing OID that is again placed under a phase for hosts that have such ORACLE_HOMEs.

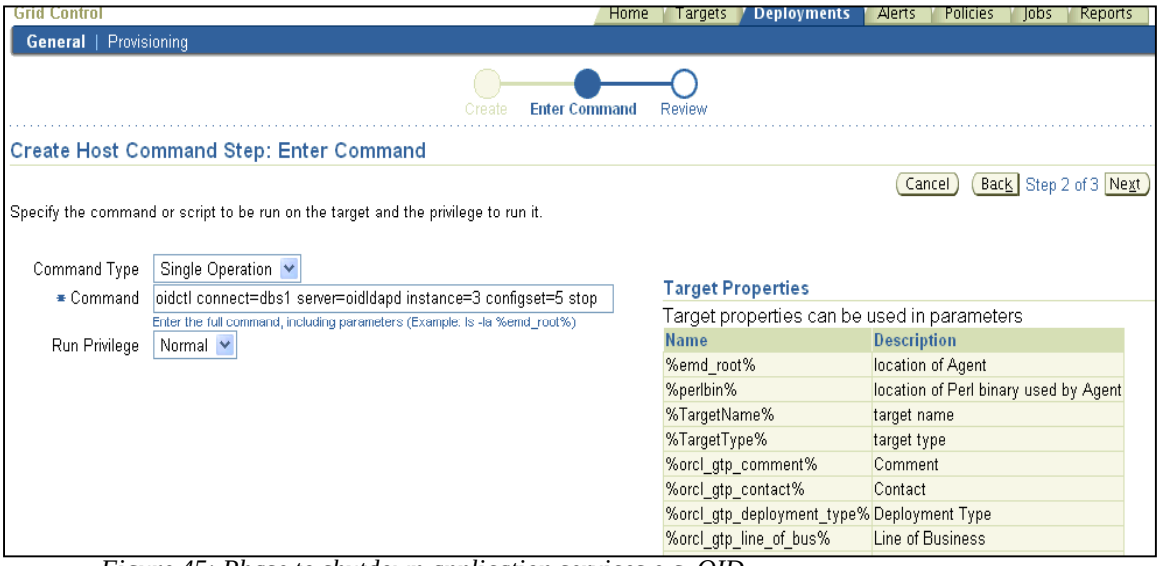


Figure 45: Phase to shutdown application services e.g. OID

Select	Name	Type	Description	Error Handling Mode
	Copy of XYZ Corp's CPU patching procedure		Procedure for applying CPUs to databases	Skip Target
	Initialize	Computational	Initializes the runtime. Do not disable or delete this step.	Inherit (Skip Target)
	Application shutdown for hosts	Parallel	Shuts down application services	Inherit (Skip Target)
	Inside Homes	Rolling	Loops over Homes	Inherit (Skip Target)
	OID shutdown	Host Command	Shuts down OID services	Inherit (Skip Target)
	For all hosts	Parallel	Iterates over a list of hosts.	Inherit (Skip Target)
	For all homes	Rolling	Iterates over a list of Oracle Homes.	Inherit (Skip Target)
	OHPreProcess	Computational	Initializes each Oracle Home. Do not disable or delete this step.	Inherit (Skip Target)
	Stage Patches	Job	Stages selected patches into Oracle Homes.	Inherit (Skip Target)
	Stop Database	Directive	Stops Oracle Database instances in an Oracle Home.	Inherit (Skip Target)
	Apply Patches	Directive	Applies staged patches in Oracle Homes.	Inherit (Skip Target)
	Start Database	Directive	Starts Oracle Database instances in an Oracle Home.	Inherit (Skip Target)

Figure 46: Host command-based step to stop OID service

c. Send notifications for the Deployment procedure run

Deployment procedures are tied up with the EM notification systems. This enables you to be notified about the status of the procedure run. Deployment procedures utilize the standard "PAF Status Notification" Rule to send out notifications. The pre-requisite is to set the methods of notifications and their setup (like SMTP server for emails) in Enterprise Manager. Also the standard rule can be customized to extend notifications to specific procedure runs and various methods of notification based on the requirement.

The steps below illustrate with examples to enable notifications and get notified on the list of status for deployment procedures. This involves enabling of Notifications in the required Deployment procedures and by configuring the Notification rule for specific jobs and methods of notification.

Customizing the Deployment Procedures for enabling Notifications:

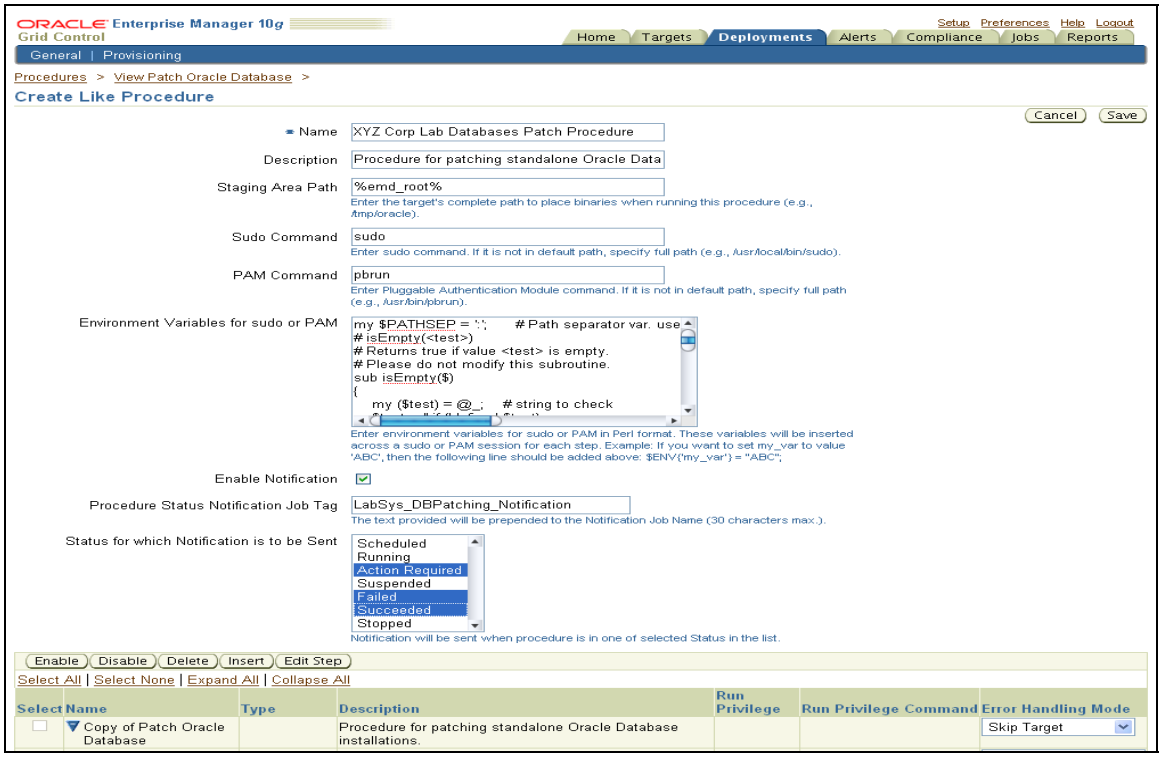


Figure 47:Enable Notification for Deployment Procedures.

1) To enable the notification, do a 'create like' the out-of-box deployment procedure. The above figure illustrates the 'XYZ Corp Lab Database Patch Procedure', which is copy of the out-of-box - "Patch Oracle Database" deployment procedure.

2) Select the check box **Enable Notification**. Provide a name for the Procedure Status Notifications Job Tag; this is to create notification rules specific to a deployment procedure. Also the job name can be associated to create a rule specific to this procedure and set notification methods as required. The figure above illustrates the Job tag specified as "LabSys_DBPatching_Notification", which is unique to the deployment procedure used to patch lab systems at XYZ Corp.

3) Select the different status from the **Status for which Notification is to be Sent** list provided. These are the statuses for which you would want to be notified for this particular Deployment procedure. In the example referred in the figure above, the status 'Action Required, Failed and Succeeded' is selected to receive the notifications for.

Configuring of Notification Rules:

The Provisioning Framework uses the out-of-box generic "PAF Status Notification" rule to send notifications for the status of the procedure run. The standard rule can simply be subscribed to send the basic Email notifications. To access the Notification rules, Click the **Preferences** link and Select the **Rules** link under the Notification section of the gun handle options in the Preferences page. The figure below shows how to enable Subscribe options for the 'PAF Status Notification' rule to send Emails. Click on **Apply** after subscribing.

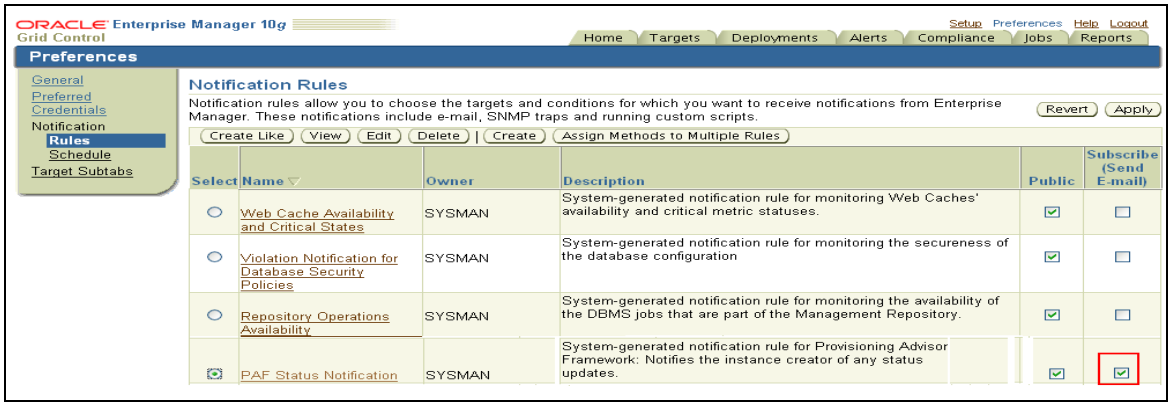


*Figure 48:*Subscribe to Emails for the Standard 'PAF Status Notification' Rule.

Advanced users can customize the standard "PAF Status Notification" rule to receive notifications in required ways for specific deployment procedures. For example, you might want to be notified by email for a test system procedure, but for a production run would want to be well informed on the status through SMS Alerts and so on. To incorporate specific requirements and enable different methods of notification, it requires editing the standard out-of-box notification rule and editing the job with the specific name and associating specific Method of notification from the pre-defined notification methods. The steps below illustrates this:

12) Do a 'Create Like' of the Standard 'PAF Status Notification' Rule. Provide a Name, for example based on the procedure you are configuring this for like 'Lab system databases patching procedure notification' rule. The figure below illustrates this.

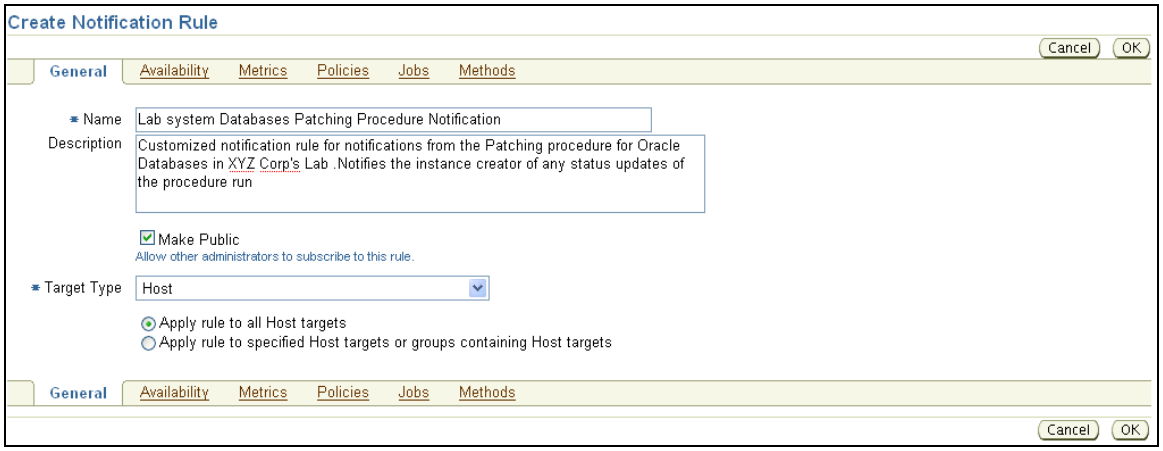


*Figure 49:"*Create Like" of Standard PAF Notification Rule and incorporating changes.

13) Select the 'Jobs' sub tab and edit the Job to associate the job name (LabSys_DBPatching_Notification) with this new notification rule. Provide the name as the job nametag used in during the customization of the deployment procedure for which notifications is enabled.

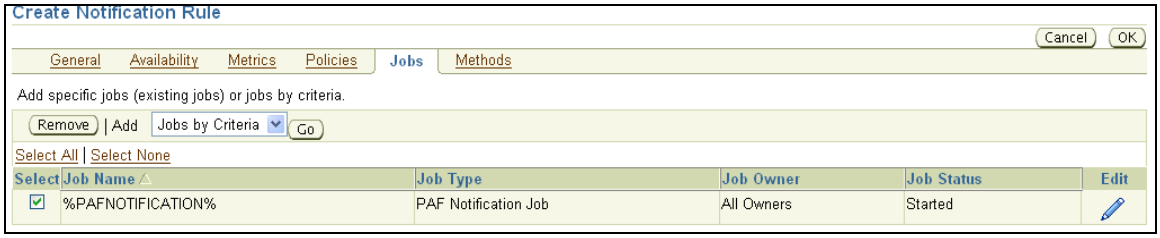


Figure 50:Edit the Job of the new PAF Status Notification Rule

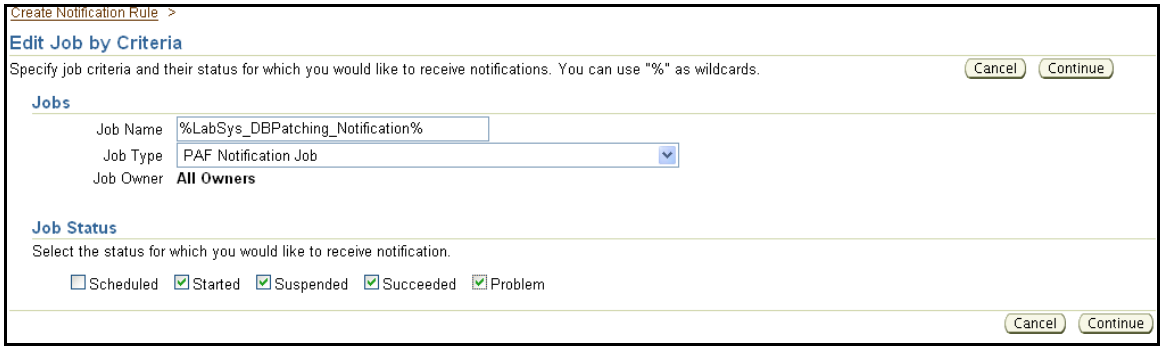


Figure 51: Associate the job name of the Custom Deployment Procedure

14) Select the “Methods” sub tab. set the required Notification Method for the new rule. Can choose the basic **Send Email** Option or pick from various Advanced Notification Method available or incorporated by you. The figure below shows the notification methods, which have been created and available with the Enterprise Manger Grid Control.

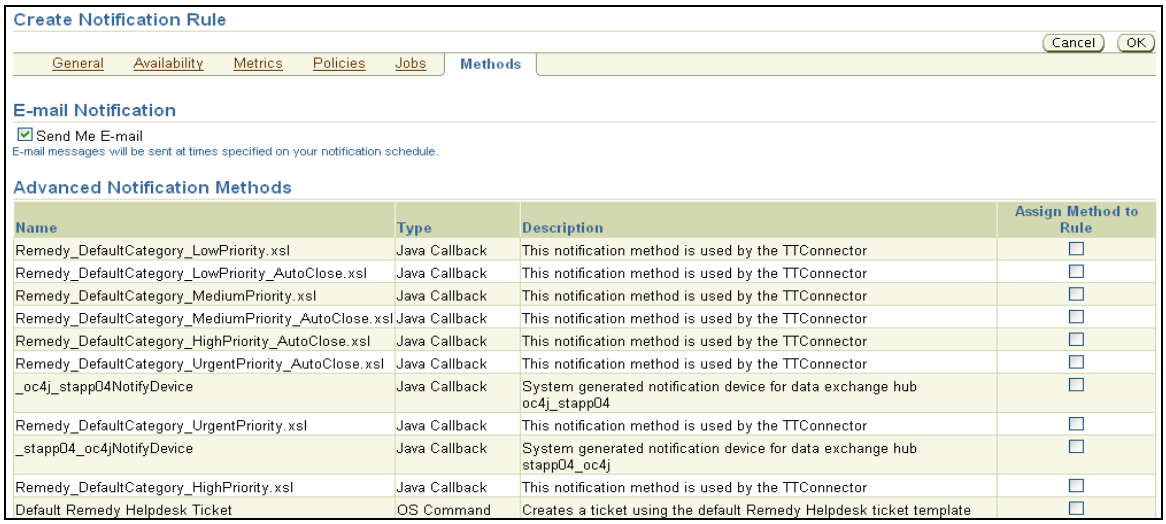


Figure 52: Set Notification Methods -Basic Email Notification and Advanced Notification Methods.

Save the new rule, this completes the process of enabling notifications and configuring rule for a specific Deployment procedure.

On executing the deployment procedure, based on the status selected under “Send Notification For Status” list the users registered with the OMS will receive notifications on various notification methods deployed.

Note: The Notification would be sent to all registered email address under the EM user account. If you want this only to a specific group or individual (for example, to the DBA group or DBA who is patching), you need to create a shared EM user or a separate EM account for the DBA and register the specific email address to Notification setup by logging into EM with the specific EM user credentials.

17. How can the out-of-box Directives be customized? If so how to do it?

Out-of-Box directive can be customized my making a copy of the Perl script associated and create it into a new directive. The use case in the following example outlines the usage.

Example for this case is: Supporting Multiple Inventory case.

Use case: I have multiple inventories in my environment. Say one for each Oracle Home. So, while patching the databases (single or a group), the inventory location file is specified.
To support the use case, the Out-of-Box (OOB) deployment procedures – Apply Patch steps needs to be edited to take up the extra parameter (invPtrLoc), which OOB results to default value.

Create the new customized directive from the OOB Directive
Step 1: Select the OOB Oracle Directive from the Software Library.
Software Library: Deployments > Provisioning > Directives > Oracle Directives >Patching >Common > All > Generic > 'Apply Patch' Directive.

Oracle Directives							Oracle Directives
BPEL Directive							BPEL Directive
Common							Common
Common Provisioning Utilities							Common Provisioning Utilities
CRS							CRS
LBR Configuration							LBR Configuration
Linux Stage Server setup							Linux Stage Server setup
myJ2EECompany Provisioning							myJ2EECompany Provisioning
NetApp Cloning							NetApp Cloning
Patching							Patching
ASM							ASM
Common							Common
All							All
Generic							Generic
Apply Patch	0.4	Ready	Untested		1.0		TestDirective description
Apply Patch 10203	0.1	Ready	Untested		1.0		Apply Patch for 10203
Apply Patch Without Logging	0.3	Ready	Untested		1.0		TestDirective description
Custom Prereq Checks	0.8	Ready	Untested		1.0		TestDirective description
Custom Shutdown DB	0.8	Ready	Untested		1.0		TestDirective description
Custom Validation Patch	0.8	Ready	Untested		1.0		TestDirective description
NApply Apply Patch	0.1	Ready	Untested		1.0		TestDirective description
PA_Apply_Post_SQL	0.7	Ready	Untested		1.0		TestDirective description
PA_Apply_SQL	0.7	Ready	Untested		1.0		TestDirective description
PA_Blackout	0.3	Ready	Untested		1.0		TestDirective description
Run_root_postscript_for_patchsets	0.4	Ready	Untested		1.0		TestDirective description
Run_the_Prereqs	0.3	Ready	Production	Oracle Database	10.2.0.3		Executes the Patch applicability prerequisite checks
Start SQL Apply	0.3	Ready	Untested		1.0		TestDirective description
Stop SQL Apply	0.3	Ready	Untested		1.0		TestDirective description
CRS							CRS
OS							OS
RAC Provisioning							RAC Provisioning

Figure 53:OOB Directive: Apply Patch in the Software Library

Step 2: Select the directive and click Edit and take a copy of the Perl script. (Open and Copy the Perl code to the local editor and save it locally.)

Best practice while doing customization is to create a ‘Custom Directives’ directory to segregate and store the custom directives. Use the **Create Folder** option to create the folder.

Figure 54: Best Practice: Create 'Custom Directives' to aggregate custom scripts

Step 3:Select the directory and click **Create Directive**.

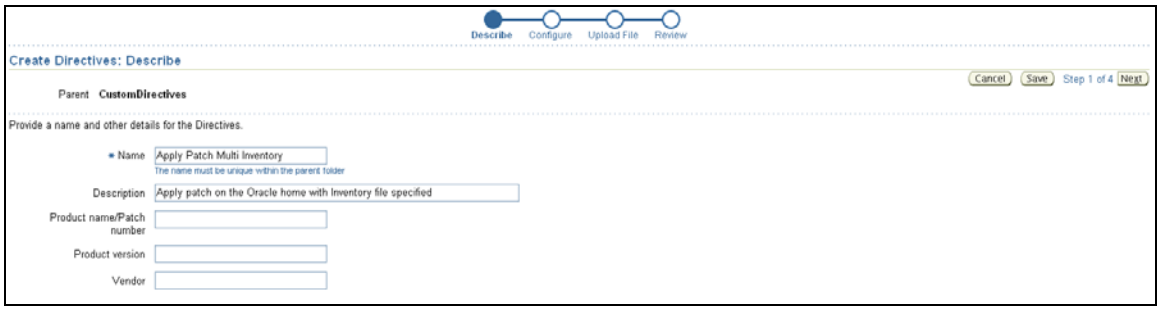


Figure 13: Create Directive 'Apply Patch Multiple Inventory'

Step 4: Click **Add** to add the Arguments required for the ‘Apply Patch Multiple Inventory’ script.

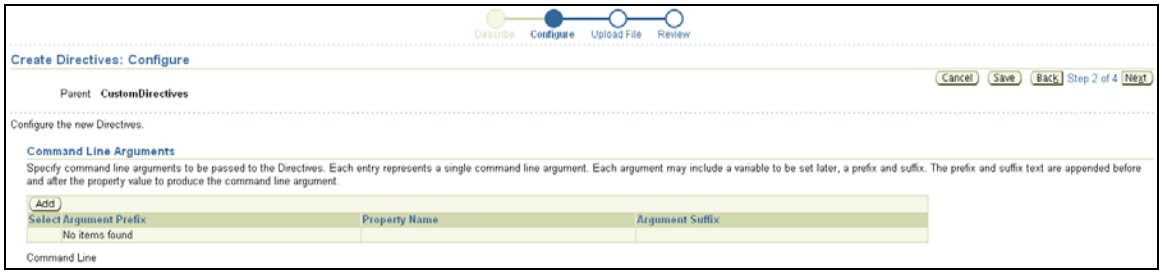


Figure 14:Add Command Line Arguments for the script.

Make sure you enter a space character after the “Argument Prefix’ for example “-invPtrLoc”

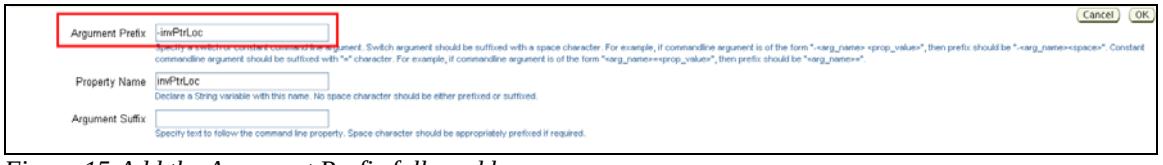


Figure 15:Add the Argument Prefix followed by a space

Repeat ‘Add’ action to add all the 5 Command line Arguments as seen in the figure below. Set the Shell Type to ‘Perl’, since using a Perl script.

Note: If you have any other script other than Perl or Bash like ‘bat’ set Shell Type to ‘Defined in Script’.

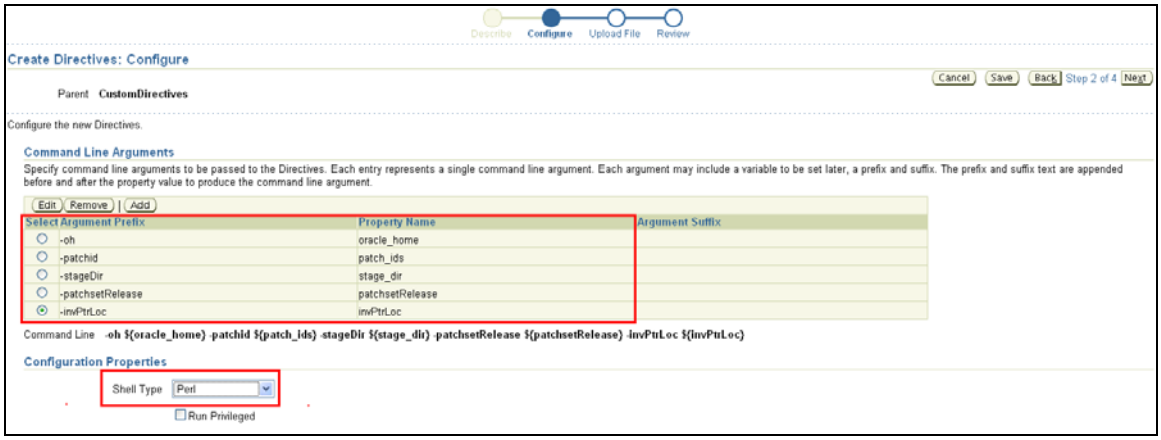


Figure 16: Command Line Argument for the Apply Patch directive

Step 5: Click **Next** and Upload the Perl script from the local system.

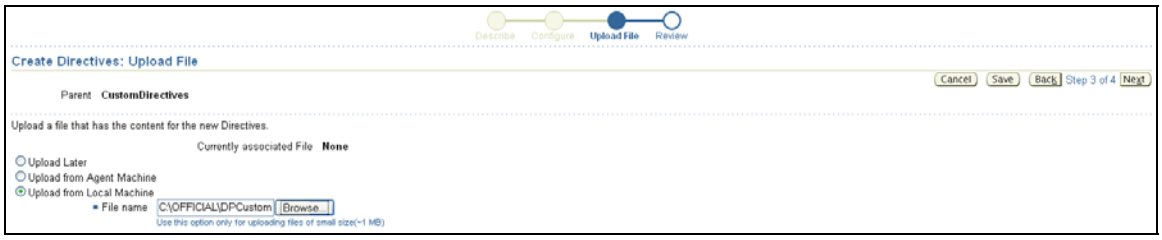


Figure 17:Upload from Local Machine - Upload the script to the Directive

Figure 18: Complete Review and Click Finish to add the new Directive

Step 6: Click **Next** and review the selection in the Review page.

Step 7: Click **Finish** and add the new directive to the Software Library.

Modify the 'Apply Patch' of the procedure to map to the new directive:

Step 1: Select the Deployment procedure 'Patch Oracle Database', click **Create Like** and edit the 'Apply Patches' to associate with the new directive created.

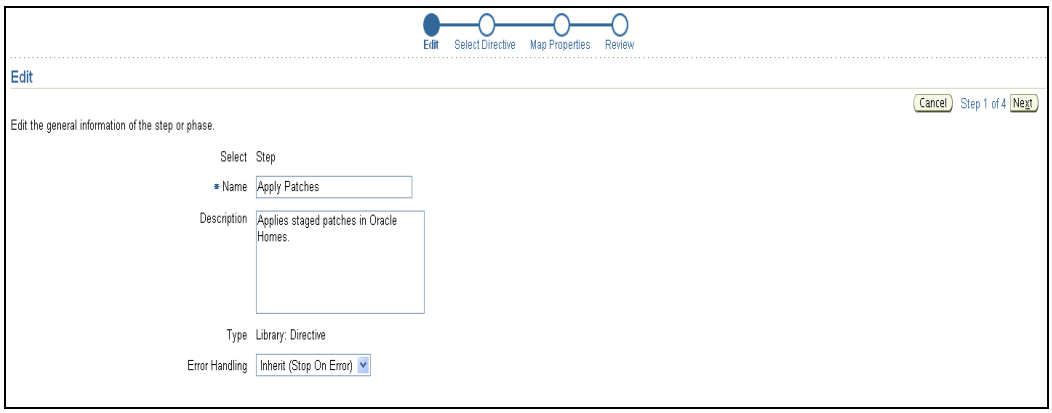


Figure 19: Edit the 'Apply Patches' step of the Procedure

Step 2: In Select Directive page, choose **Select New Directive** and pick the new directive created. Make sure you select the **Always Use Latest Revision** check box to associate the step to the latest directive.

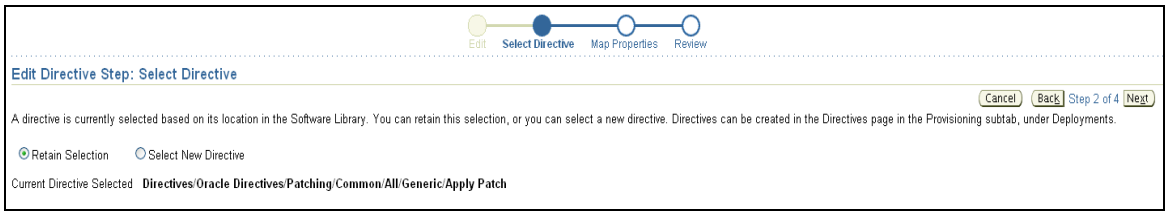


Figure 620: Check 'Select New Directive' to pick the newly created directive.

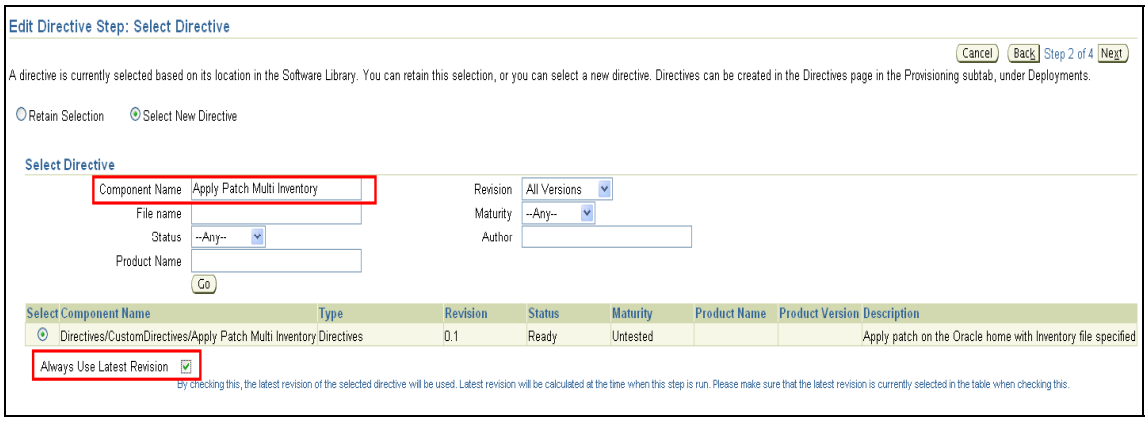


Figure 21: Select the new directive and check 'Always Use Latest Review '.

Step 3: Click **Next** and Map the properties for the arguments as seen in the figure below.

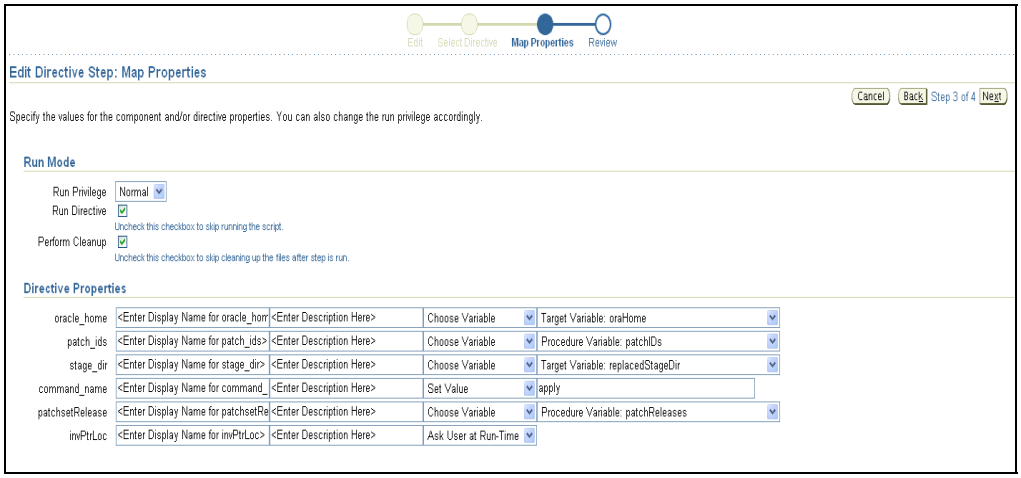


Figure 22: Map the properties for the Directive Arguments.

Note: If your inventory pointer location is a standard location for your environment across targets, you can map that property to **Set Value** and to the location in your environment. If it is not a common one and you would want to specify during the inventory based on the target groups. Make the property selection dynamic by setting it to ‘Ask User at Run Time’ as seen in the figure above.

Step 4: Click **Next**, review the selection and click **Finish** to complete editing the step.

Step 5: Save the Procedure with a unique name (say Patch Oracle Database with Multiple Inventory). Select and click **Schedule Deployment**.

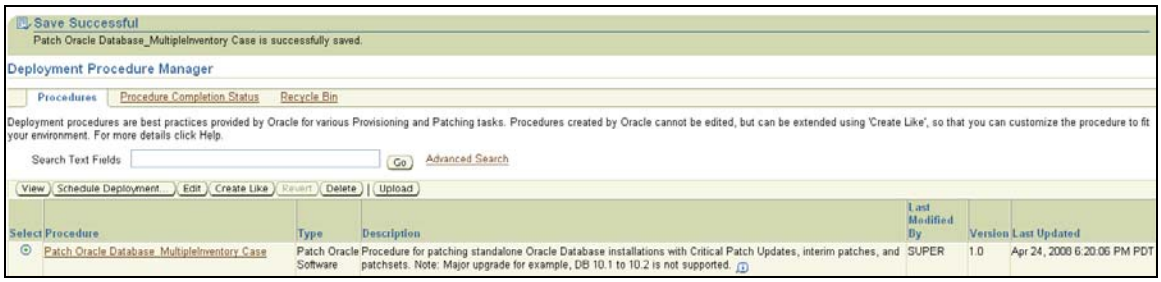


Figure 23: Save the procedure and use for the Patching Databases with non-default inventory.

During the interview:
Specify the Inventory Pointer location in the Library Step Properties page during the interview. Click through and submit the procedure for execution and monitor the status of execution from the Procedure Completion Status page.

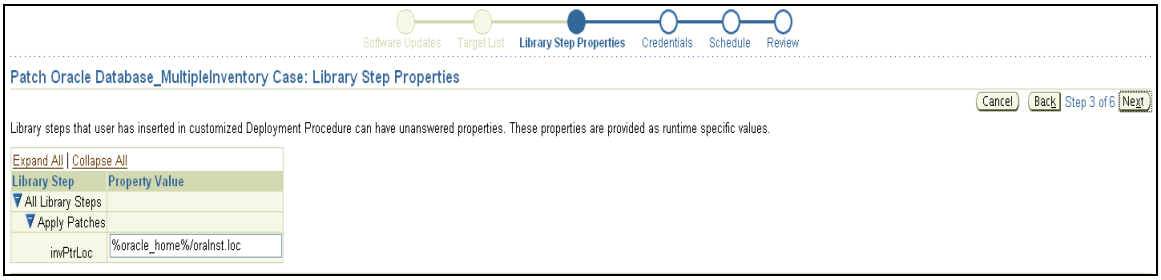


Figure 24: Specify the Inventory Pointer location for the targets chosen.

Note: RAC Procedures support: The same can be applied to the RAC Procedures and the same Apply Patch directive created can be used. So for using in RAC Case, make a create like of the RAC Procedure, and edit the 'Apply Patches' step to uptake the new directive.

18. Can Deployment Procedures be executed using a Pluggable Authentication Module (PAM) and sudo?

Yes. Deployment Procedures allow execution in sudo and other PAM contexts. The sudo and the PAM command to be used throughout the procedure can be specified while editing the procedure (see figure 58). The individual directive steps can be chosen to run in normal mode, privileged mode (which requires sudo setup at the target for superuser execution for the command) or as a different user than the preferred credentials. (If its EM 10.2.0.3, apply the one-off patch (5890474) on the OMS Oracle Home).

The **use case** is described below in *italics*:

The various actors (Operating System users on the target box) involved are:

1. *Oracle: This is the Database Home owner. This is a locked account.*
2. *emd: This is the Agent Home owner.*
3. *foo: This is the user who is performing the patching operation.*

These are just examples. The user names in actual environments may vary.

Requirements for the various users are as follows:

1. *All the three users oracle, emd and foo belong to the same group, which is the OS DBA group.*
2. *User foo can sudo to user oracle using foo's password.*
3. *In the 'sudoers' file of the Target, foo should have access to "<AgentHome>/perl/bin/perl"*

The goal is to patch the Database Home using foo's credentials.

Design Time experience (one time):

This process needs to be done just once to customize the Deployment Procedure.

1. *User selects the Oracle shipped Patch Oracle Database Deployment Procedure and does a Create Like for that Deployment Procedure and gives it some name.*
2. *User selects run-as sudo for all steps in the edited Deployment Procedure.*
3. *User selects the sudo command to be run. In this example it may be "sudo –u oracle"*
4. *User gives the staging location for the directives in "Staging Area Path". The default is %oracle_spool%/EMStagedPatches/Others. Example: It could be %emd_root%/EMStagedPatches/Others This directory needs to have 770 permissions. This could also be any other directory, which can be written onto by user "oracle".*
5. *User saves the modified Deployment Procedure (Patch Oracle Database with sudo).*

Run Time experience (every time):

This process needs to be done for every run of the Deployment Procedure

User selects the modified patching Deployment Procedure (Patch Oracle Database with sudo) and runs it.

1. *User selects the Database Updates to be patched.*
2. *User provides the location for staging the patch. The default for this is: User gives the PA Staged location as: %oracle_spool%/EMStagedPatches/Others*
   Example: It could be %emd_root%/EMStagedPatches/Others This directory needs to have 770 permissions and it should exist before the DP is run. This could also be any other directory, which can be written onto by user "oracle".
3. *Now user gives foo's credentials (Patching User credentials)*
4. *Schedule and submit the job.*

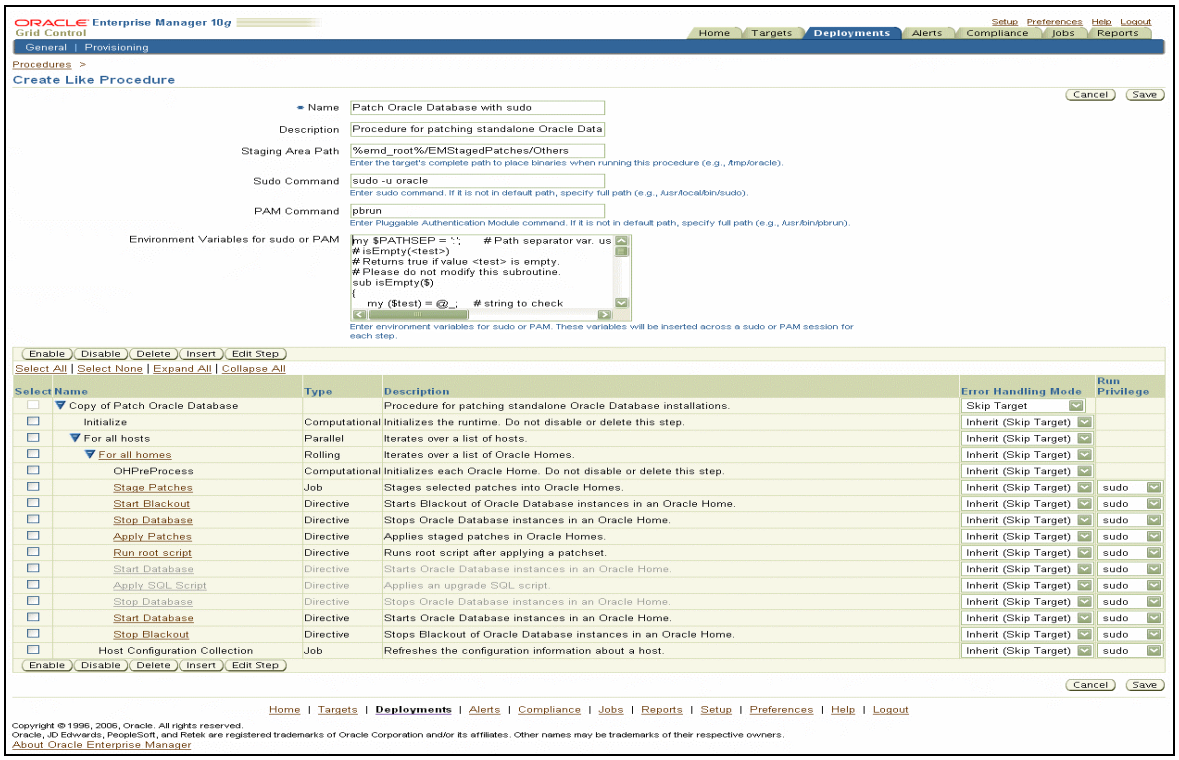


Figure 67: Screenshot for modifying a Deployment Procedure for sudo

19. How can we back-up the deployment procedures? Or transfer them from one Grid Control environment to another?

In version 10.2.0.3 there is an out-of-box PAR Deploy utility to export deployment procedures created by a user and distribute them across various Oracle Enterprise Manager deployments.

Provisioning Archive files can contain deployment procedures and/or components and directives from Software Library. If you have created a custom deployment procedure, which you would want to export to another home, you would require creating a PAR file containing the procedure and exporting it from the OMS. The same file can be imported or deployed on another OMS either from the UI or through the command line.

Refer to Section 10.4.2 – Importing or Exporting Deployment Procedures under "Enterprise Manager Advanced Configuration" from the library list at

http://download-west.oracle.com/docs/cd/B16240_01/doc/nav/portal_booklist.htm

20. What are the out-of-box variables exposed with deployment procedures?

Enterprise Manager exposes several variables that can be used with Deployment Procedures. These variables can be used by our customers to customize their specific tasks like startup, shutdown, etc using their own directives.

Database specific:

oraHome - directory of the ORACLE_HOME
instances - selected database targets from the ORACLE_HOME
all_instances_home- all database targets running from the ORACLE_HOME
dbSIDs - all sids from the ORACLE_HOME
dbListeners - all listeners from the ORACLE_HOME
runRootScript - yes/no indicating if rootscript needs to be run

Automated Storage Management specific:

asmTargetHome – selected ASM target ORACLE_HOME
asmInstanceName – ASM instance running from the ORACLE_HOME
asminstances – ASM instances running from the ORRACLE_HOME

Real Application Cluster specific:

racLocalInstnaceNames :
All local RAC Database instances running out of the ORACLE_HOME
racLocalInstanceTgtNames :
All local RAC Database Target names running out of the ORACLE_HOME
RacLocalInstnaceHomes :
All ORACLE_HOMES running the RAC database instances locally.
racLocalInstanceSids : SIDs of the local RAC instances running in the ORACLE_HOME

Clusterware specific:

nodeName : Name of CRS node on which the RAC instance being patched is running.
crsName : Name of the Cluster.

Application Server specific:

oracleSid : Value of SID that may be present in an AS ORACLE_HOME

Global:

isPatchset - yes/no of if patchset is being applied to the ORACLE_HOME
replacedStageDir: The absolute staging location of patches
patchIDs: List of patch ids selected
patchSrcs: Indicating whether the patches came from *Metalink* or software lib
patchData :Uniform resource Names (URNs) of the patches
patchReleases : Corresponding release of the patches.

targetVersion : Version of the target being patched.
Note: stageDir - the staging directory to use provided like %oracle_home% (Use replacedStageDir)

21.Is there a CLI runtime for Deployment Procedures?

Yes, the Deployment Procedures can be run from Command Line Interface. The basic CLI verb 'emcli submit_procedure' is enhanced with the scripts, which can be utilized to run deployment procedures. This supports command line execution of the deployment procedures out-of-box and customized deployment procedure with option to schedule, scale and customize the run time values dynamically.

Make sure the pre-requisite patch – 5890474 is applied to the OMS 10.2.0.3 and pick the latest EMCLI executables and templates from the OMS Home.

For further details refer to Section 10: Using EMCLI to execute deployment procedures under "Enterprise Manager Advanced Configuration" guide from the library list at
http://download-west.oracle.com/docs/cd/B16240_01/doc/nav/portal_booklist.htm

22. Which log files are to be looked for in case of any errors while executing the deployment procedures? Can these be associated with the SR's?

The Deployment procedure logs in OMS Home and the Agent Home of the Target.

OMS Side logs:

Generic EM trace file - $OMS ORACLE_HOME /sysman/log/emoms.trc

PAF logs - $OMS ORACLE_HOME/ sysman/log/pafLogs/

For specific DP instance there exists a log inside:

$OMS_ORACLE_HOME/sysman/log/pafLogs/<name>_<instance_guid>.log

Agent Side Logs:

$Agent_ORACLE_HOME/sysma/logs/emagent.nohup

$Agent_ORACLE_HOME/sysma/logs/emagent.trc

Advanced Options:

Optionally, to capture more details you can make the logging finer. Follow the steps below to re-set the log level and capture the logs mentioned above. (Note: Its advised to archive the old logs and have a fresh run after resetting the log level to capture the fresh logs.)

In the OMS Side:

"$ORACLE_HOME/sysman/config/emomslogging.properties"file

@ log4j.rootCategory=....

Replace the value of the above parameter to 'DEBUG'. Bounce the OMS for the changes to take effect:

OMS Home/bin/emctl stop oms

OMS Home/bin/emctl start oms

In the Agent side of the Target:

AGENT_HOME/sysman/config/emd.properties

tracelevel.Dispatcherr=DEBUG (Writes to emagent.nohup)

tracelevel.command=DEBUG (Writes to emagent.trc)

Re-load the agent: $Agent_ORACLE_HOME/bin/emctl reload agent

The settings above are to be set only when you want to extra details and when the logs don't have sufficient information to debug the issue. Make sure to set the debug level back to the original levels after reproducing the issue.

While reporting issues with deployment procedures, associate the tar/zip of the logs from both the above locations with the SR.

23. Can I solve conflicts while applying NApply patches using Deployment Procedures? If so, how?

The standard patching Deployment Procedures intelligently identify if a given patch is a normal patch or napply and calls the corresponding command to apply the patch.(Either apply/napply).
The procedures by default trigger napply with OPatch option: skip_subset. (Skips the molecules from applying if it is a subset of any patch in the OH)
The OPatch also has other options like:

skip_duplicate: Skip the molecules while applying, which are duplicates of that in the Oracle Home.

skip_conflict: Skip the molecules while applying , which are conflicting with any patches in the Oracle Home.

skip_missing_components: Skip the molecules while applying, which have some dependant components, which are missing, in the patch or in the Oracle Home.

In case of conflicts during applying a patch, the application of the patch will fail and the logs would explain the reason behind the failure. In that case, one could use any or all of the above OPatch options to successfully execute the patch application through the procedures.
(This can be adopted by default but watch out for the logs of the Apply Patch step).

All the above options are already available as a part of the Perl directive code, which is commented out by default. Uncomment the lines to be used as a part of the procedure. The procedure to skip the errors/conflicts during patching and apply the rest of the molecules/patches is described below:

. The directive “Apply Patch 10203” has to be edited to enable this. Do the following:
   a. Go to “Deployments->Provisioning->Directives”
   b. Select> Directives/OracleDirectives/Patching/Common/All/Generic/ApplyPatch and click **Edit**
   c. Go to Upload File section.
   d. Download the file “pa_patch_oraclehome.pl” from here.
   e. Edit it as given below and upload it back by selecting **Upload from Local Machine**.

The changes to be done to the directive are:
Go to line 1370 and removes the comment and put the comment to 1371.
Similarly remove the comment on line 1397 and put it on 1398.

1370,1371 – Before
#. " $skip_conflict $skip_missing_comp $skip_subset 2>&1");
. " $skip_subset 2>&1");

1370,1371 – After
. " $skip_conflict $skip_missing_comp $skip_subset 2>&1");
#. " $skip_subset 2>&1");

The existing flow for applying CPUs is illustrated in the following screenwatch:
http://download.oracle.com/enterprise_manager/10203_CPU_Patching/10203_CPU_Patching.html

24. Can I provision Single Instance Databases using deployment procedures? If so, how?

One can provision Databases from either a reference Installation or from a gold image in the Software library. This can be done through the procedure " Oracle Database Provisioning" under the page 'Provisioning Procedures' from 'Deployments'
The example below shows the steps to provision Databases using the 'Oracle Database Provisioning' procedure.

1) Select the procedure "Oracle Database Provisioning" from the Deployment Procedures page under Deployments tab. (See figure below).

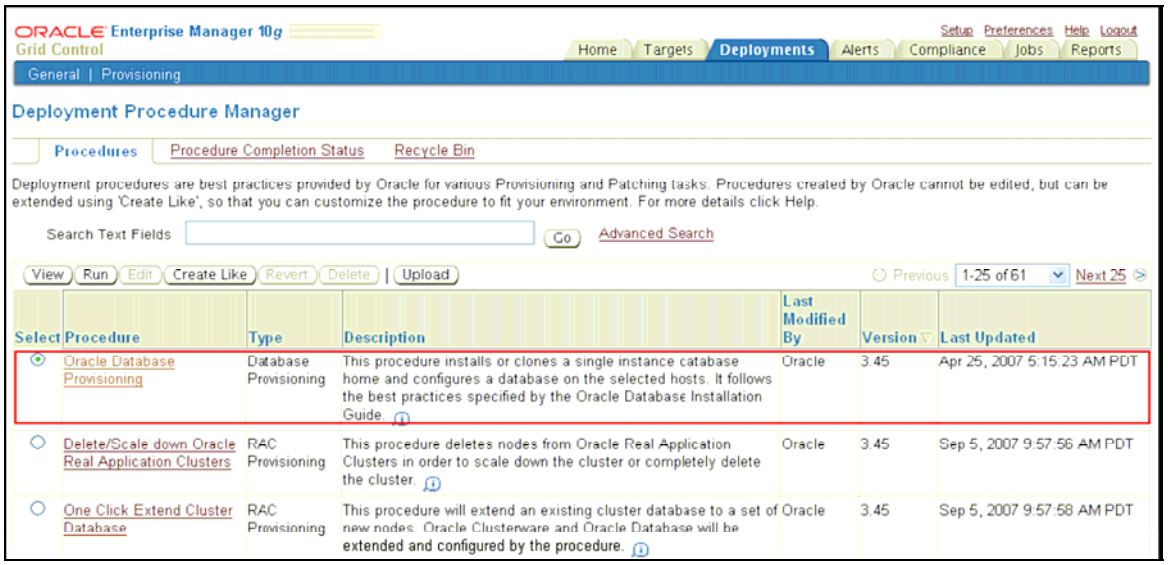


Figure 68: 'Oracle Database Provisioning' procedures.

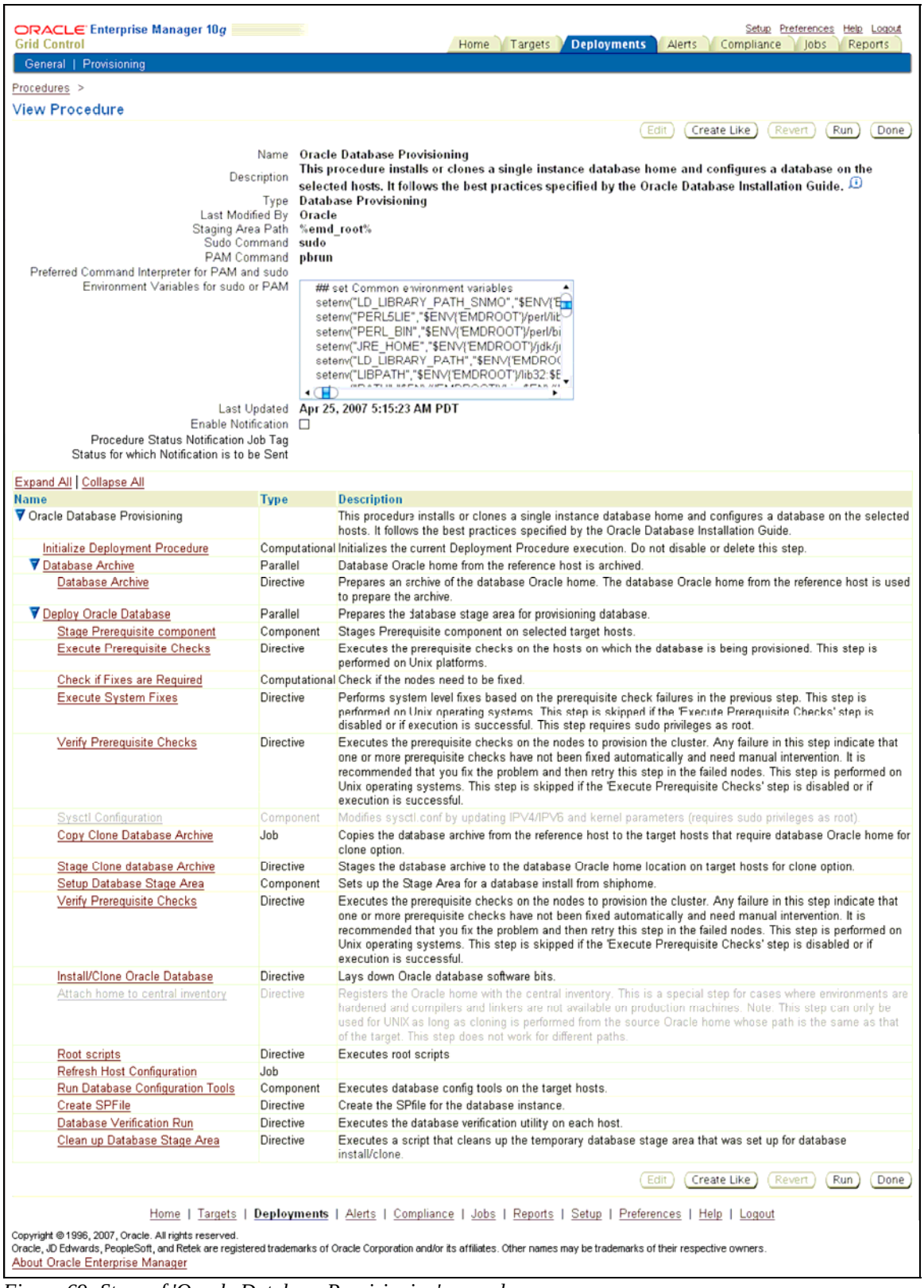


Figure 69: Steps of 'Oracle Database Provisioning' procedure.

2) Click on **Run** to start the procedure. The source of can be selected either from the Installed Home or Software Library. In this example we will use the database clone component in software library to provision. (See figure 59 and 60 below)

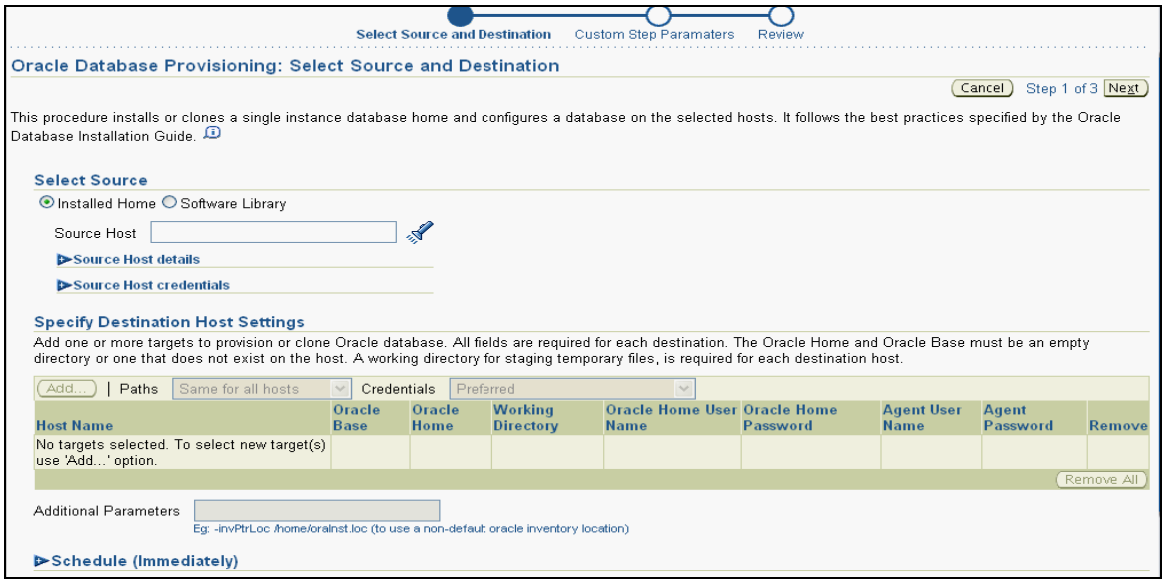


Figure 70: Source Selection: Installed Home or Software Library.

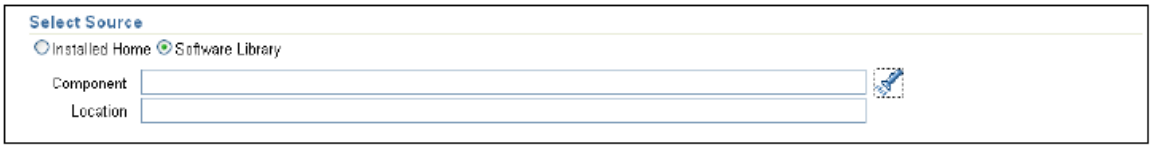


Figure 71: Select the Source from Software Library.

3) Select the pre-staged database clone component from the Software library. (See figure below)

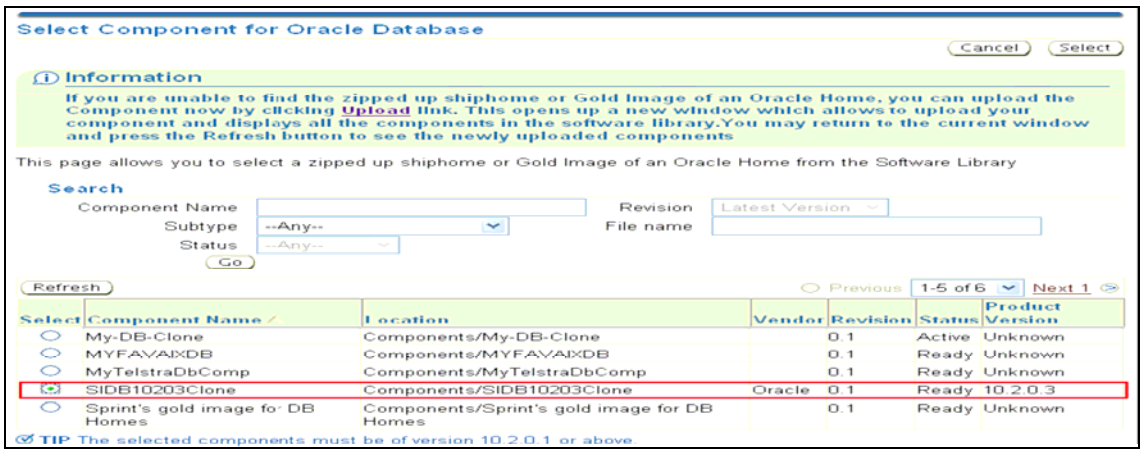


Figure 72: Select "SIDB10203Clone" -10.2.0.3 DB Clone Component from Software Library

4) Proceed to select the Destination Target for provisioning the database. The Enterprise Manager shows up the pre-selected target list, which matches the criteria of the Source selected. (See Figure #62 below).

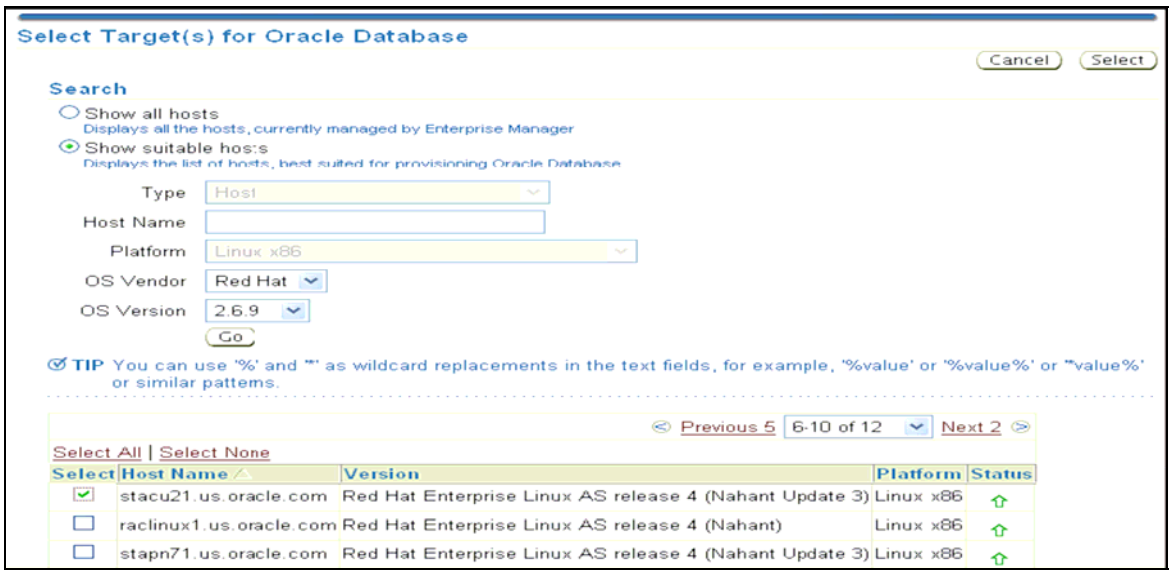


Figure 73: Pre-populated destination target list as per the Source selected.

5) Fill in the details of Oracle Base and Oracle Home on the Destination target. By default the values for Oracle Base and Homes are populated as per Oracle best practices, override for your usage. (See figure #63 below)

6) By default Credentials are set to 'Preferred Credentials'. If not stored, select to override. In this example 'Override-Same for all hosts' option is selected and filled in the details of the Home and Agent credentials.

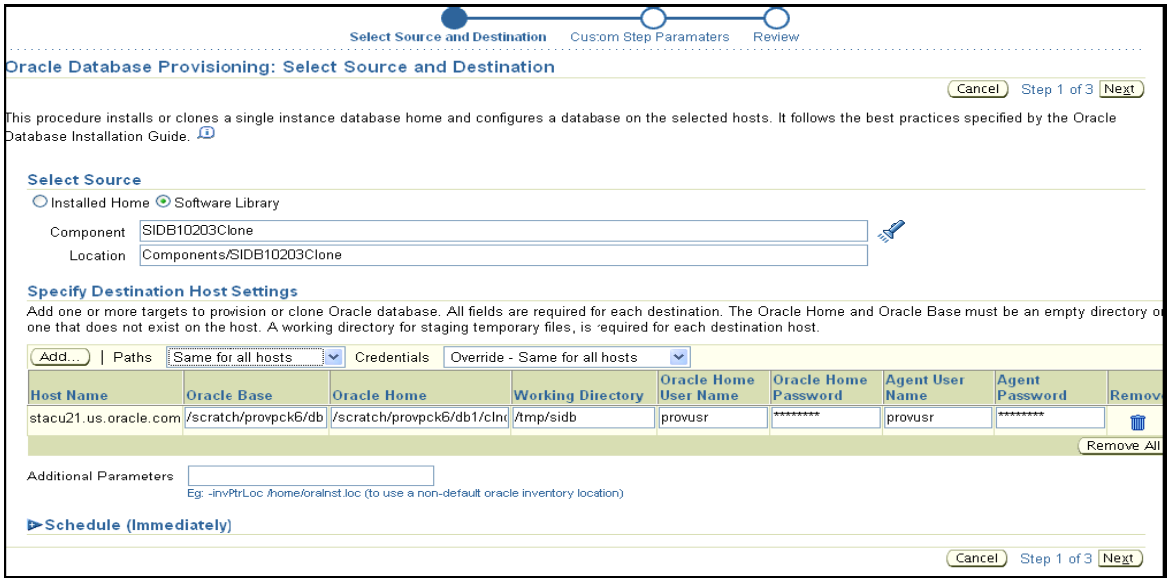


Figure 74: Provide in details on the Destination Target.

7) Click **Next** to proceed on the 'Custom Step Parameters' step. Fill in values attributes asked for.(See Figure #64 below).

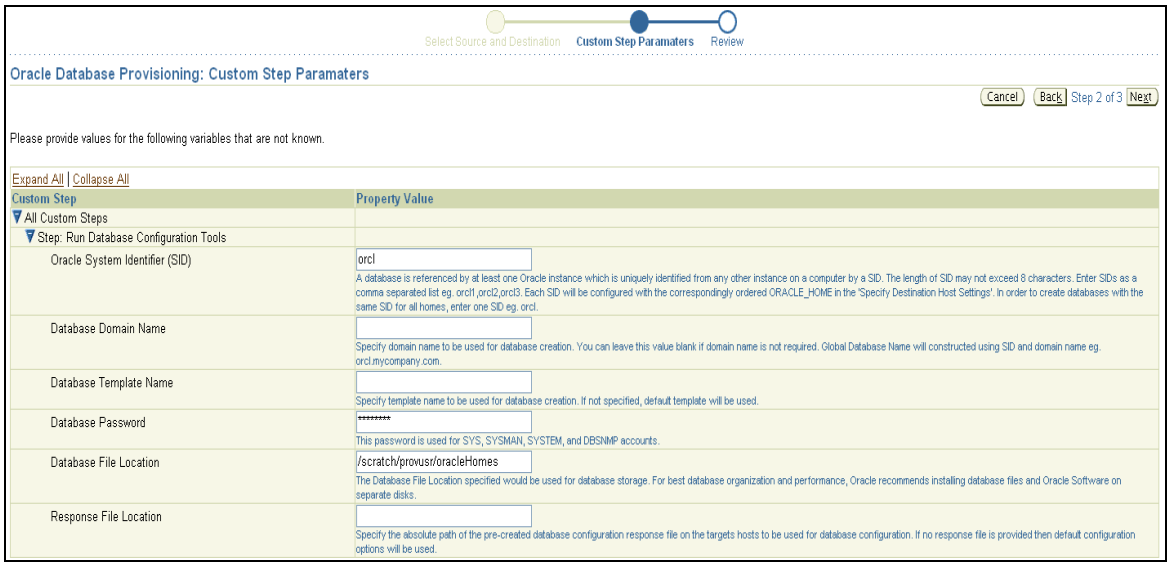


Figure 75: Fill in values for the Custom Step Parameters.

Database password is applicable to accounts SYS, SYSMAN, SYSTEM, and DBSNMP.
Mandatory parameters: 'Oracle System Identifier (SID)', 'Database password',' Database File Location',

Optional parameters:
Database Domain Name, Database Template Name (If you want a template based Provisioning), and Response File Location: If not specified, the default configuration options will be chosen for the provisioning.

Note: For installing multiple SIDB instances provide comma-separated list of SID's.

8) Click **Next** and proceed to Review the options selected. (See figure below)

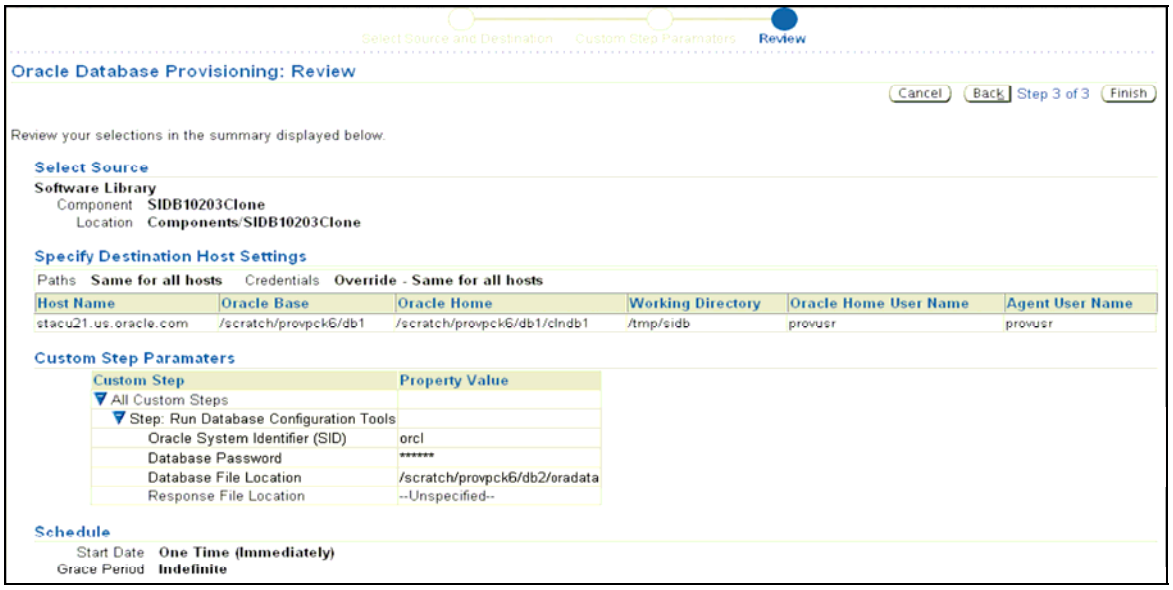


Figure 76:Oracle Database Provisioning-Review the Selection

9) Click **Finish** to execute the procedure and monitor the steps from the Procedure Completion Status page.

On completion of the execution, there will be a new Database provisioned on the destination target.

For Oracle Home Software Clone only:
If you want only to do Cloning operation, do a create like of the 'Oracle Database Provisioning' procedure and selectively disable steps of Database creation and save as a custom procedure. (See figures below)

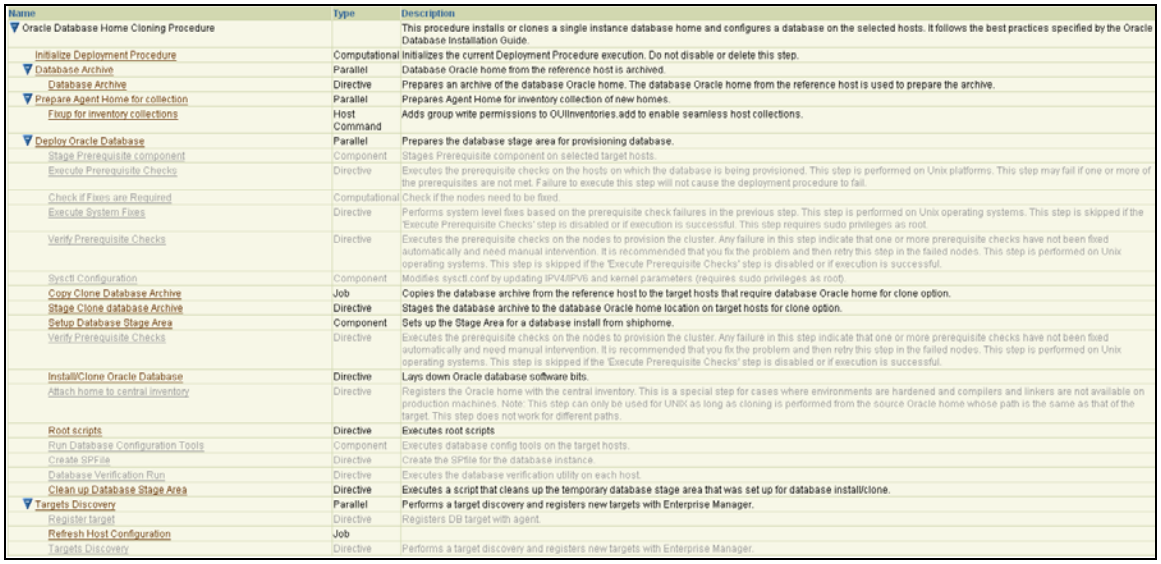


Figure 77: Cloning Procedure: Created from 'Oracle Database Provisioning' procedure.

25. Can I apply patchsets say 10.2.0.4 on 10.2.0.2 DB using Deployment Procedures?

To apply patchset upgrades say 10.2.0.4 on top of 10.2.0.3/10.2.0.2 or 10.2.0.1 you can use the standard 'Patch Oracle Database' procedures with just enabling the 'Run Root Script step.

Do 'Create like' of the standard procedure and enable the 'Run Root Script' and save the procedure assay 'Patch Oracle Database (Patchsets)'.

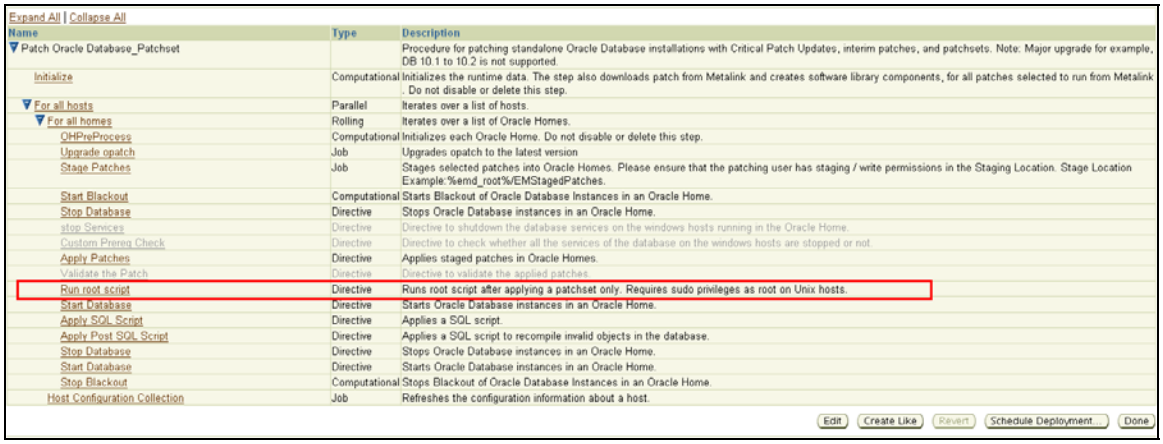


Figure 78: Enable 'Run Root Script' step to apply DB patchsets.

Note: On enabling 'Run Root Script' step, set the step to execute in SUDO or PAM (say pbrun, suexec) mode for example: sudo –u root or pbrun root

Support: The procedure supports minor upgrades of database like 9.2.0.x to 9.2.0.X, 10.1.0.1 to 10.1.0.X, and 10.2.0.1 to 10.2.0.X (X=2,3,or 4) of a version of a database. Deployment Procedures does not support upgrading major version like 10.1.0.x to 10.2.0.x

26. Can I Provision or Patch 11g Databases using Deployment Procedures? Does it require any certification patch to be applied to the OMS?

Yes, 11g DB can be Provisioned and Patched using Deployment Procedures.

To provision 11g DB: Use the standard provisioning procedures like 'Provision Oracle Database' procedure. You do not require any certification patch over the OMS of EM 10.2.0.4

To patch 11g DB: With CPU and One-off patches.

Apply the patch '6058648' on the Oracle Enterprise Manager (OMS_ORACLE_HOME) to get the support and use the standard procedure 'Patch Oracle Database'.

Technical document

April 2008
Authors: Sudip Datta, and Hariprasanna Srinivasan
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200