Database patching through Enterprise Manager 10*g*-An Illustrated Discourse

Sudip Datta, Oracle Corporation.

Table of Contents

1 Introduction

Patching is a complex and time-consuming exercise in today's datacenters. According to a recent Aberdeen group study, patch handling costs businesses in excess of 2 billion dollars annually. For a leading service provider, the cost was reported to be as high as $14,400 per server.

Prior to 10*g* release, Database Administrators (DBAs) would need to look up Metalink, Oracle's official knowledge repository for customers, for pertinent patches have to manually download, stage and apply them. The DBA would have to repeat the patch application over a set of nodes and often over multiple ORACLE_HOMEs per node. This not only is cumbersome but also non-scalable as implement a Grid of large number of distributed yet similar nodes. Furthermore, the whole manual process is time consuming leading to poor Quality of Service (QOS) and higher Total Cost of Ownership (TCO).

Currently, Oracle releases software in three different forms:

A full software release commonly referred to as the *Base release*.

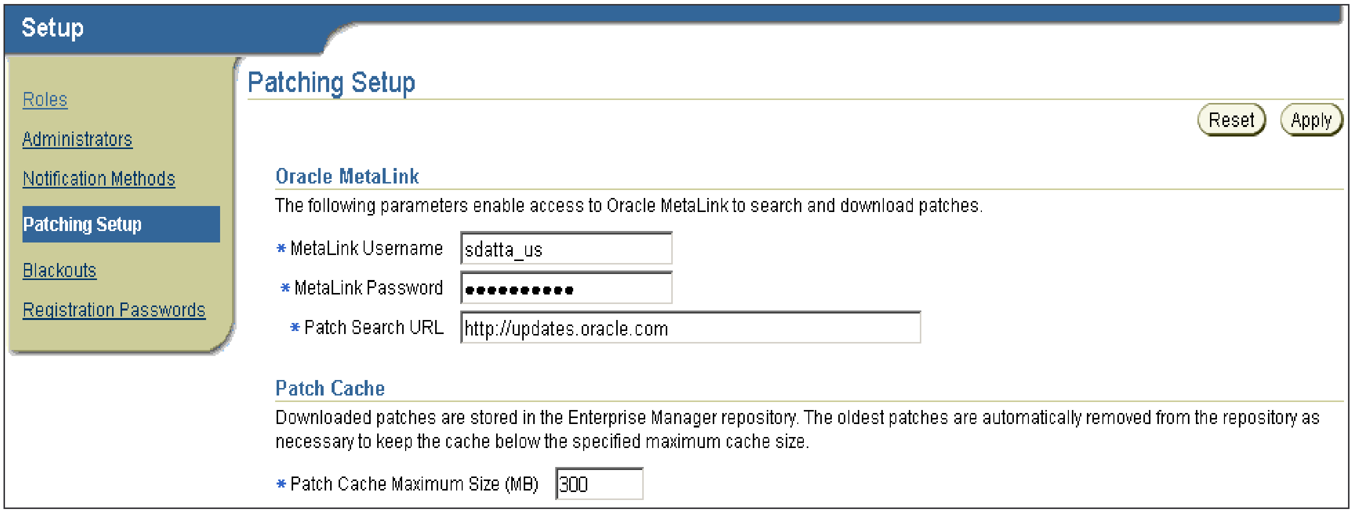
A cumulative software release commonly referred to as *Patchsets*.

Incremental bug fixes commonly referred to as *Interim Patches*.

Oracle Enterprise Manager 10*g* makes it easy for customers to track interim patches of their interest and apply them to the software. It supports a live connection to Oracle Metalink and can refresh periodically to reflect the availability of patches. This proactive notification would be extremely beneficial in cases of security and corruption alerts where customers would need to apply the patches as soon as they become available.

2 Accessing metalink from Enterprise manager

Enterprise Manager can be set up to refresh the patch list periodically from Metalink. The set up involves Metalink credentials and connection details. In most customer places, the Enterprise Manager is installed well within the corporate firewall. In those cases, the connection to the internet is not be a direct one, but via a proxy. The Metalink credentials and the connection can be configured both out out-of-box and subsequently via Enterprise Manager. One can either enter the details during the Grid Control install or by using the Setup tab of the Grid Control. The patches when downloaded can be staged in a patch cache.



Patching Setup

Oracle MetaLink

The following parameters enable access to Oracle MetaLink to search and download patches.

* MetaLink Username: sdatta_us

* MetaLink Password: ●●●●●●●●●●●

* Patch Search URL: http://updates.oracle.com

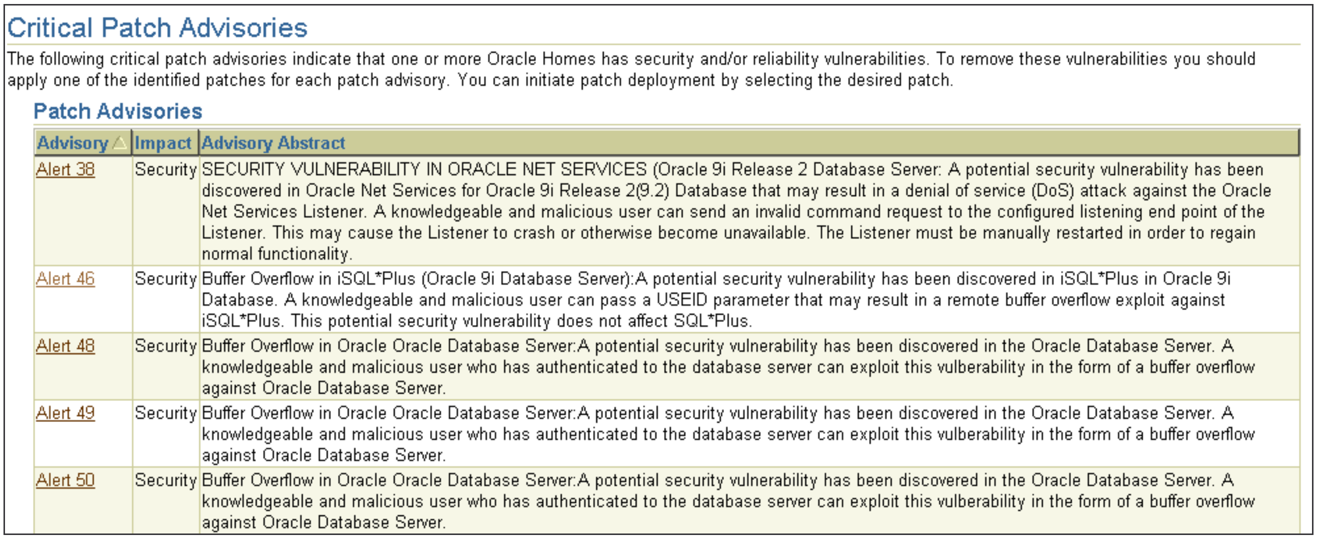
Patch Cache

Downloaded patches are stored in the Enterprise Manager repository. The oldest patches are automatically removed from the repository as necessary to keep the cache below the specified maximum cache size.

* Patch Cache Maximum Size (MB): 300

3 CHOOSING THE RIGHT PATCH

With Enterprise Manager 10g Oracle has extended its proactive services to the desk of a DBA. The Enterprise Manager now keeps a watchful eye on any critical alerts that arrive via Metalink. The alerts may be related to sensitive areas like Security, data integrity or availability. This means that customers no longer have to wait before they hit a critical bug, rather they are notified as soon as Oracle has developed a fix for the problem. The Critical patch advisory is also chained with the Patch Wizard, which enables to patch all affected targets in the enterprise in a single go. This avoids error prone operations like searching Metalink with the right keywords, getting the right patch for the platform and a manual apply.

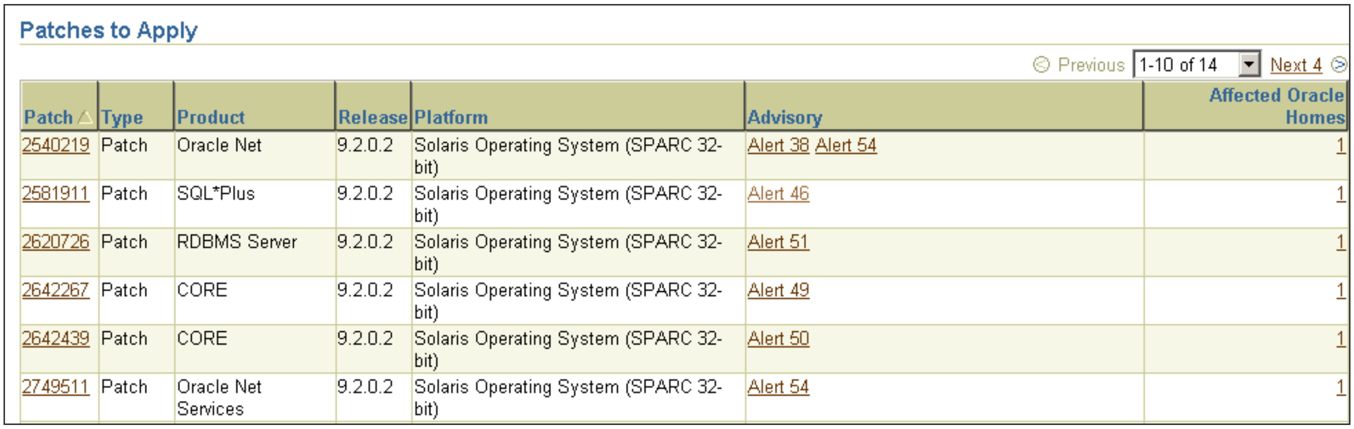


Critical Patch Advisories

The following critical patch advisories indicate that one or more Oracle Homes has security and/or reliability vulnerabilities. To remove these vulnerabilities you should apply one of the identified patches for each patch advisory. You can initiate patch deployment by selecting the desired patch.

Patch Advisories

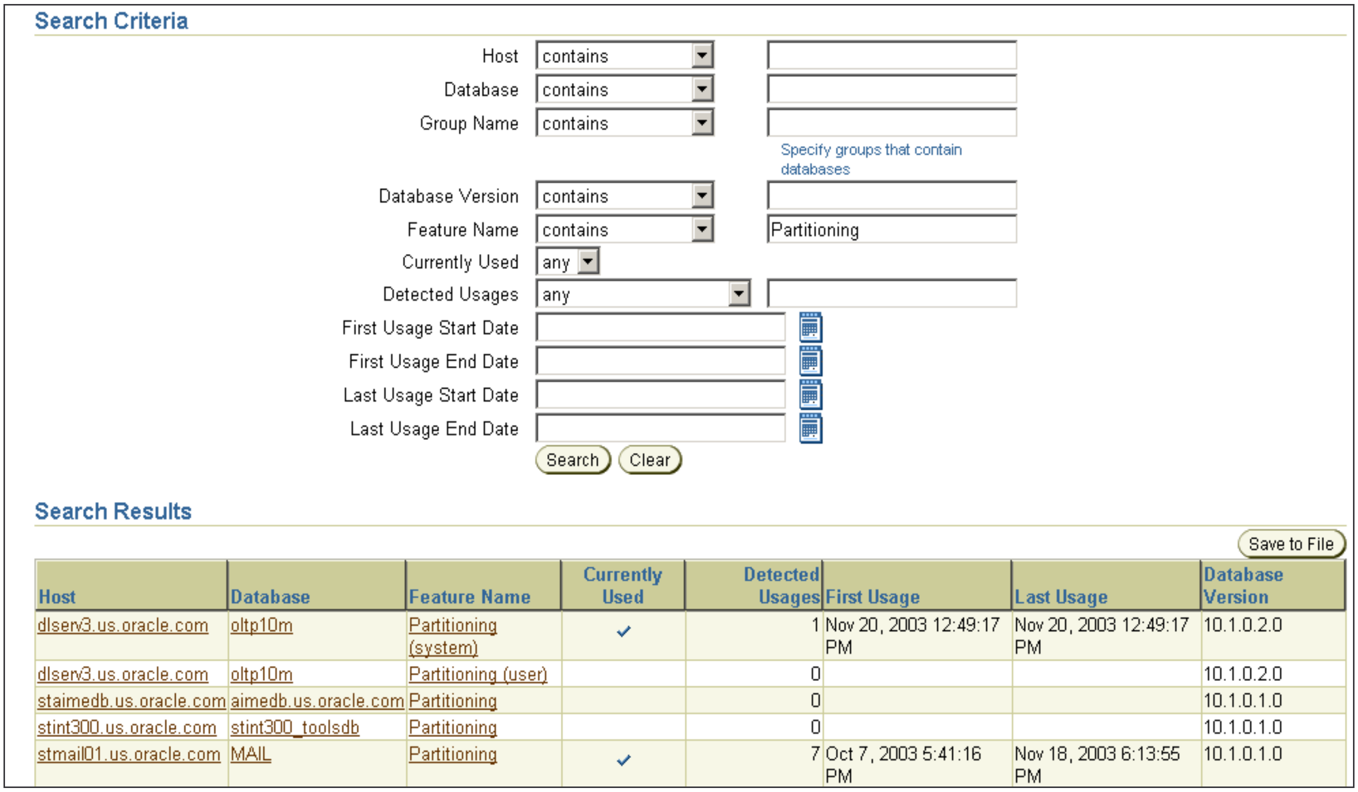
Advisory △	Impact	Advisory Abstract
Alert 38	Security	SECURITY VULNERABILITY IN ORACLE NET SERVICES (Oracle 9i Release 2 Database Server: A potential security vulnerability has been discovered in Oracle Net Services for Oracle 9i Release 2(9.2) Database that may result in a denial of service (DoS) attack against the Oracle Net Services Listener. A knowledgeable and malicious user can send an invalid command request to the configured listening end point of the Listener. This may cause the Listener to crash or otherwise become unavailable. The Listener must be manually restarted in order to regain normal functionality.
Alert 46	Security	Buffer Overflow in iSQL*Plus (Oracle 9i Database Server):A potential security vulnerability has been discovered in iSQL*Plus in Oracle 9i Database. A knowledgeable and malicious user can pass a USEID parameter that may result in a remote buffer overflow exploit against iSQL*Plus. This potential security vulnerability does not affect SQL*Plus.
Alert 48	Security	Buffer Overflow in Oracle Oracle Database Server:A potential security vulnerability has been discovered in the Oracle Database Server. A knowledgeable and malicious user who has authenticated to the database server can exploit this vulberability in the form of a buffer overflow against Oracle Database Server.
Alert 49	Security	Buffer Overflow in Oracle Oracle Database Server:A potential security vulnerability has been discovered in the Oracle Database Server. A knowledgeable and malicious user who has authenticated to the database server can exploit this vulberability in the form of a buffer overflow against Oracle Database Server.
Alert 50	Security	Buffer Overflow in Oracle Oracle Database Server:A potential security vulnerability has been discovered in the Oracle Database Server. A knowledgeable and malicious user who has authenticated to the database server can exploit this vulberability in the form of a buffer overflow against Oracle Database Server.



Patches to Apply

Patch △	Type	Product	Release	Platform	Advisory	Affected Oracle Homes
2540219	Patch	Oracle Net	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 38 Alert 54	1
2581911	Patch	SQL*Plus	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 46	1
2620726	Patch	RDBMS Server	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 51	1
2642267	Patch	CORE	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 49	1
2642439	Patch	CORE	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 50	1
2749511	Patch	Oracle Net Services	9.2.0.2	Solaris Operating System (SPARC 32-bit)	Alert 54	1

Furthermore, the new Configuration Pack also has powerful "Search" capabilities to filter out alerts that are inapplicable to the targets within the enterprise. For example, if a particular alert pertains to the "Partitioning" feature of the database, one can do a simple search to find out the targets across the enterprise that are likely to be affected by this bug. In the example below, the search yields that five databases have the partitioning feature installed and two are currently using it. The DBA can then selectively apply the patch to the right databases.



Search Criteria

Host	contains	
Database	contains	
Group Name	contains	
		Specify groups that contain databases
Database Version	contains	
Feature Name	contains	Partitioning
Currently Used	any	
Detected Usages	any	
First Usage Start Date		
First Usage End Date		
Last Usage Start Date		
Last Usage End Date		

Search Results

Host	Database	Feature Name	Currently Used	Detected Usages	First Usage	Last Usage	Database Version
dlserv3.us.oracle.com	oltp10m	Partitioning (system)	✓	1	Nov 20, 2003 12:49:17 PM	Nov 20, 2003 12:49:17 PM	10.1.0.2.0
dlserv3.us.oracle.com	oltp10m	Partitioning (user)		0			10.1.0.2.0
staimedb.us.oracle.com	aimedb.us.oracle.com	Partitioning		0			10.1.0.1.0
stint300.us.oracle.com	stint300_toolsdb	Partitioning		0			10.1.0.1.0
stmail01.us.oracle.com	MAIL	Partitioning	✓	7	Oct 7, 2003 5:41:16 PM	Nov 18, 2003 6:13:55 PM	10.1.0.1.0

Apart from this proactive notification facility, Enterprise Manager also allows the DBAs to search for specific patches based on advice from Oracle Support or the Application provider. This has been covered in the section 4.1.1.

4 The anatomy of Patching

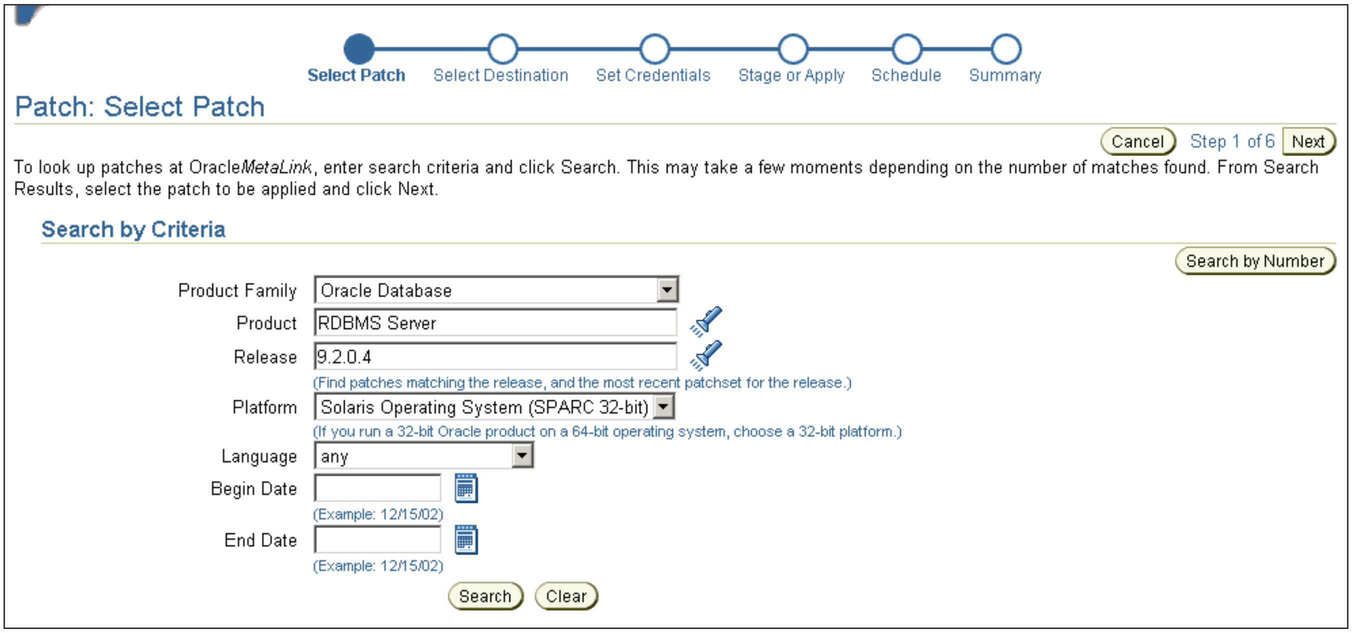
The patching via Enterprise Manager consists of a front-end application known as Patch Wizard. The back-end consists of the Job Subsystem and a standalone utility called opatch in the target ORACLE_HOME acting as the patch engine.

4.1 Patch Wizard

The Patch Wizard is a step-by-step procedure for selecting and submitting a Patch Job. It consists of the following steps:

4.1.1 Selecting a patch

The Patch selection page consists of a Search that allows searching the Metalink with criteria that suits the target to be patched. By default, the search is pre-populated with the version of Oracle and the Operating system flavor. One can also use the “Begin Date” and the “End Date” criteria to find out patches released within a specific period. Once the list of patches shows up, one can select a patch for application to a list of targets that satisfy the applicability.



Patch: Select Patch

To look up patches at Oracle*MetaLink*, enter search criteria and click Search. This may take a few moments depending on the number of matches found. From Search Results, select the patch to be applied and click Next.

Search by Criteria

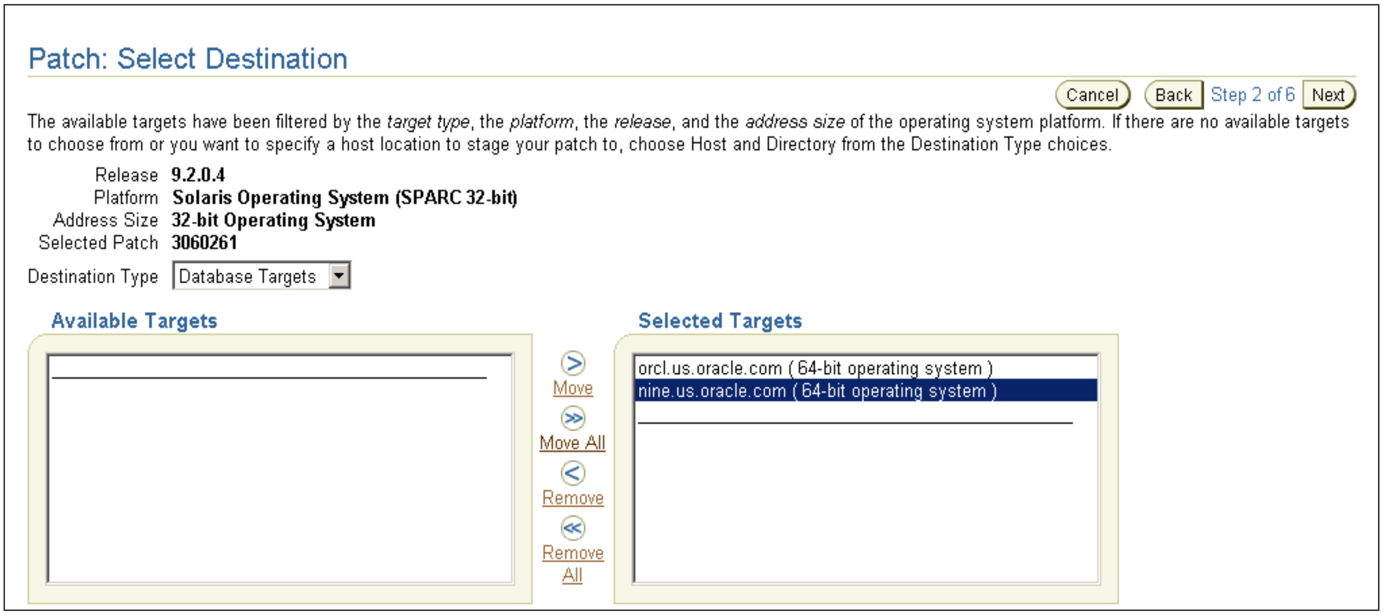
Product Family	Oracle Database
Product	RDBMS Server
Release	9.2.0.4 (Find patches matching the release, and the most recent patchset for the release.)
Platform	Solaris Operating System (SPARC 32-bit) (If you run a 32-bit Oracle product on a 64-bit operating system, choose a 32-bit platform.)
Language	any
Begin Date	(Example: 12/15/02)
End Date	(Example: 12/15/02)

4.1.1 Selecting one or more targets

The list of targets will display one or more targets that satisfy the criteria for the patch. The criteria includes the product name (Database, iAS etc), product version and the Operating system. One can select a list of targets or take off specific targets from the list if it is not desired to patch a specific target (a specific example would be a database target which does not have Intermedia installed may not be included in the list for an Intermedia patch).

If the one wants to patch an ORACLE_HOME that does not have targets associated yet, one has to use the “Host and Directory” Destination type and specify host and the location. A typical case would be a patch released for creating the database instance itself.

One can patch multiple ORACLE_HOMEs residing across multiple hosts in this manner.



Patch: Select Destination

The available targets have been filtered by the *target type*, the *platform*, the *release*, and the *address size* of the operating system platform. If there are no available targets to choose from or you want to specify a host location to stage your patch to, choose Host and Directory from the Destination Type choices.

Release	**9.2.0.4**
Platform	**Solaris Operating System (SPARC 32-bit)**
Address Size	**32-bit Operating System**
Selected Patch	**3060261**
Destination Type	Database Targets

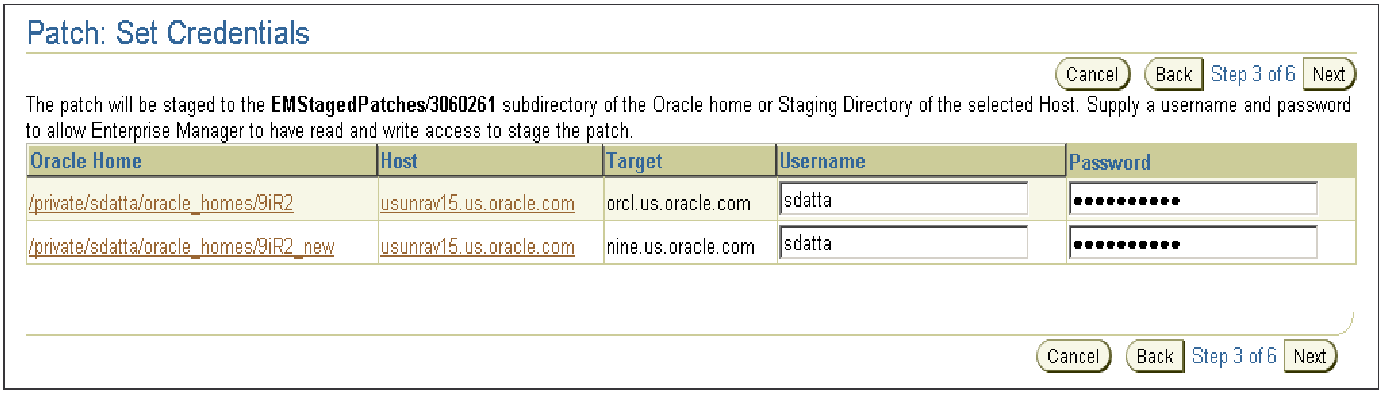
Available Targets

Selected Targets

orcl.us.oracle.com (64-bit operating system)
nine.us.oracle.com (64-bit operating system)

4.1.2 Specifying credentials

Once the targets have been specified one will need to provide the credentials for running the patch job against that. The page will have an entry for each ORACLE_HOME being patched. For each of the ORACLE_HOMEs, one needs to provide the credentials to run the job and in turn update the inventory as well as the ORACLE_HOME behind the scenes. The username and password is stored as in the repository and will get automatically populated whenever the patch wizard is used subsequently.



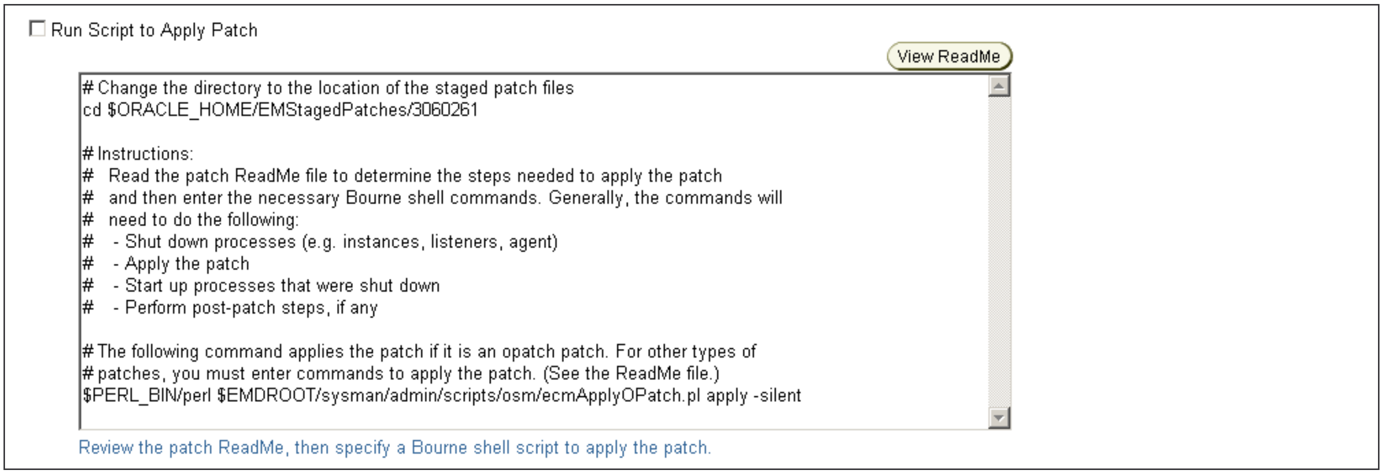
Patch: Set Credentials

The patch will be staged to the **EMStagedPatches/3060261** subdirectory of the Oracle home or Staging Directory of the selected Host. Supply a username and password to allow Enterprise Manager to have read and write access to stage the patch.

Oracle Home	Host	Target	Username	Password
/private/sdatta/oracle_homes/9iR2	usunrav15.us.oracle.com	orcl.us.oracle.com	sdatta	●●●●●●●●●●●
/private/sdatta/oracle_homes/9iR2_new	usunrav15.us.oracle.com	nine.us.oracle.com	sdatta	●●●●●●●●●●●

4.1.3 Staging or applying the patch

The Patch wizard helps in staging or applying the patch depending upon the specific requirement. With every Oracle patch there is a README file, which carries patch specific and environment specific instructions. At this point, the default behavior does not take into consideration the environment specific or patch specific variations and hence does not handle it automatically. One needs to follow the README and make changes in the default script. A few environment specific cases have been treated in the section 6.



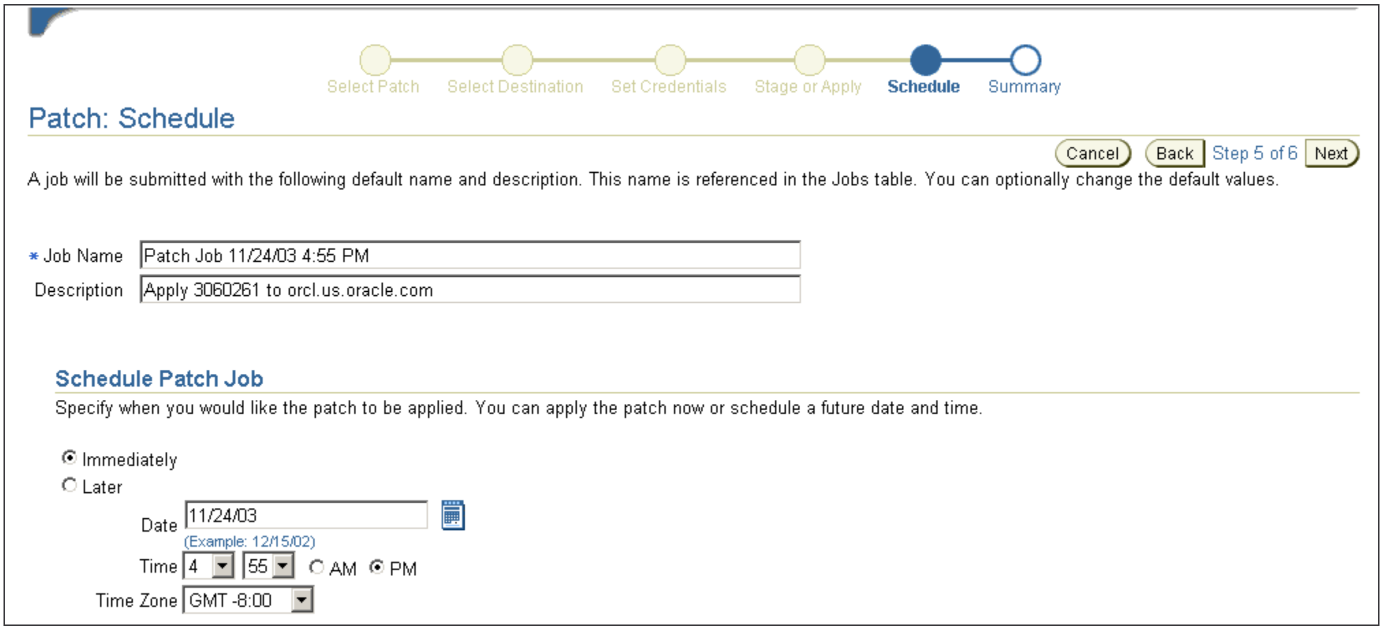
□ Run Script to Apply Patch

```
# Change the directory to the location of the staged patch files
cd $ORACLE_HOME/EMStagedPatches/3060261

# Instructions:
#   Read the patch ReadMe file to determine the steps needed to apply the patch
#   and then enter the necessary Bourne shell commands. Generally, the commands will
#   need to do the following:
#    - Shut down processes (e.g. instances, listeners, agent)
#    - Apply the patch
#    - Start up processes that were shut down
#    - Perform post-patch steps, if any

# The following command applies the patch if it is an opatch patch. For other types of
# patches, you must enter commands to apply the patch. (See the ReadMe file.)
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent
```

Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

4.1.4 Scheduling a patch application

After the script has been finalized, a job needs to be scheduled for patch application. The job can be scheduled for immediate or a later execution. The scheduler supports different time zones to handle cases where the Enterprise Manager Console and the target may lie in different geographies. The Job Summary screen is then displayed for confirmation. Once the job is submitted, it can be monitored from the Jobs tab.



Patch: Schedule

A job will be submitted with the following default name and description. This name is referenced in the Jobs table. You can optionally change the default values.

* Job Name: Patch Job 11/24/03 4:55 PM

Description: Apply 3060261 to orcl.us.oracle.com

Schedule Patch Job

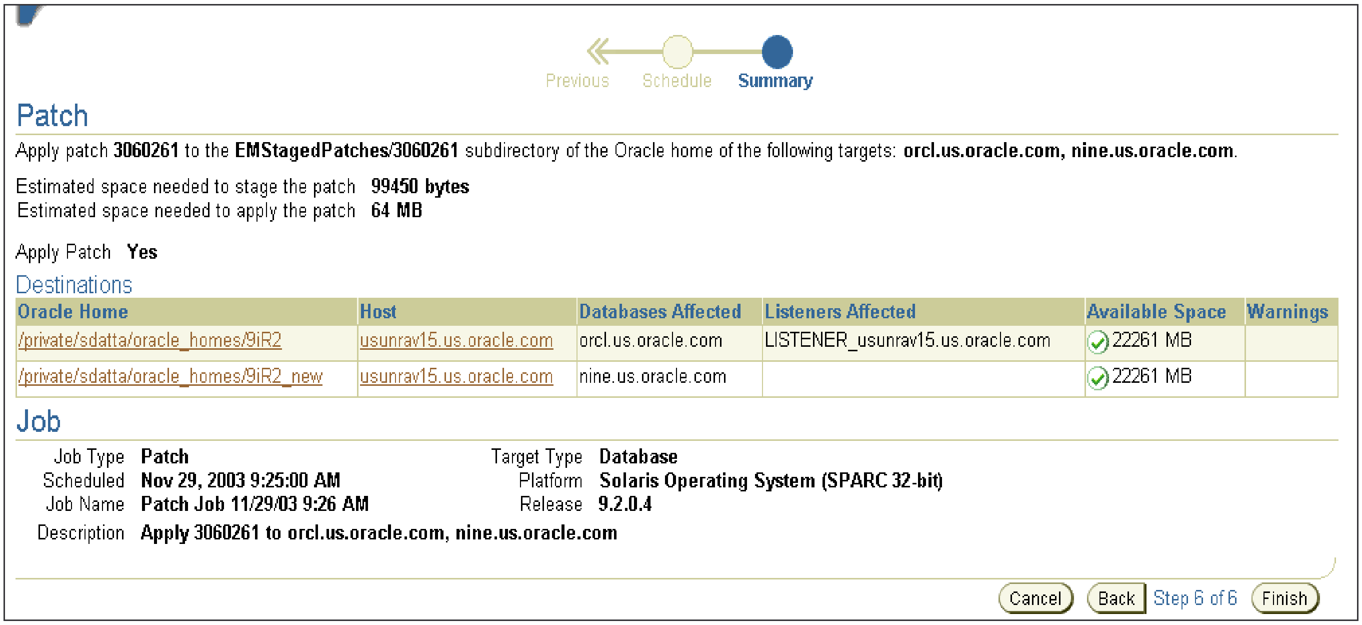
Specify when you would like the patch to be applied. You can apply the patch now or schedule a future date and time.

● Immediately
○ Later

Date: 11/24/03
(Example: 12/15/02)

Time: 4 : 55 ○ AM ● PM

Time Zone: GMT -8:00



Patch

Apply patch **3060261** to the **EMStagedPatches/3060261** subdirectory of the Oracle home of the following targets: **orcl.us.oracle.com, nine.us.oracle.com**.

Estimated space needed to stage the patch **99450 bytes**
Estimated space needed to apply the patch **64 MB**

Apply Patch **Yes**

Destinations

Oracle Home	Host	Databases Affected	Listeners Affected	Available Space	Warnings
/private/sdatta/oracle_homes/9iR2	usunrav15.us.oracle.com	orcl.us.oracle.com	LISTENER_usunrav15.us.oracle.com	✓ 22261 MB	
/private/sdatta/oracle_homes/9iR2_new	usunrav15.us.oracle.com	nine.us.oracle.com		✓ 22261 MB	

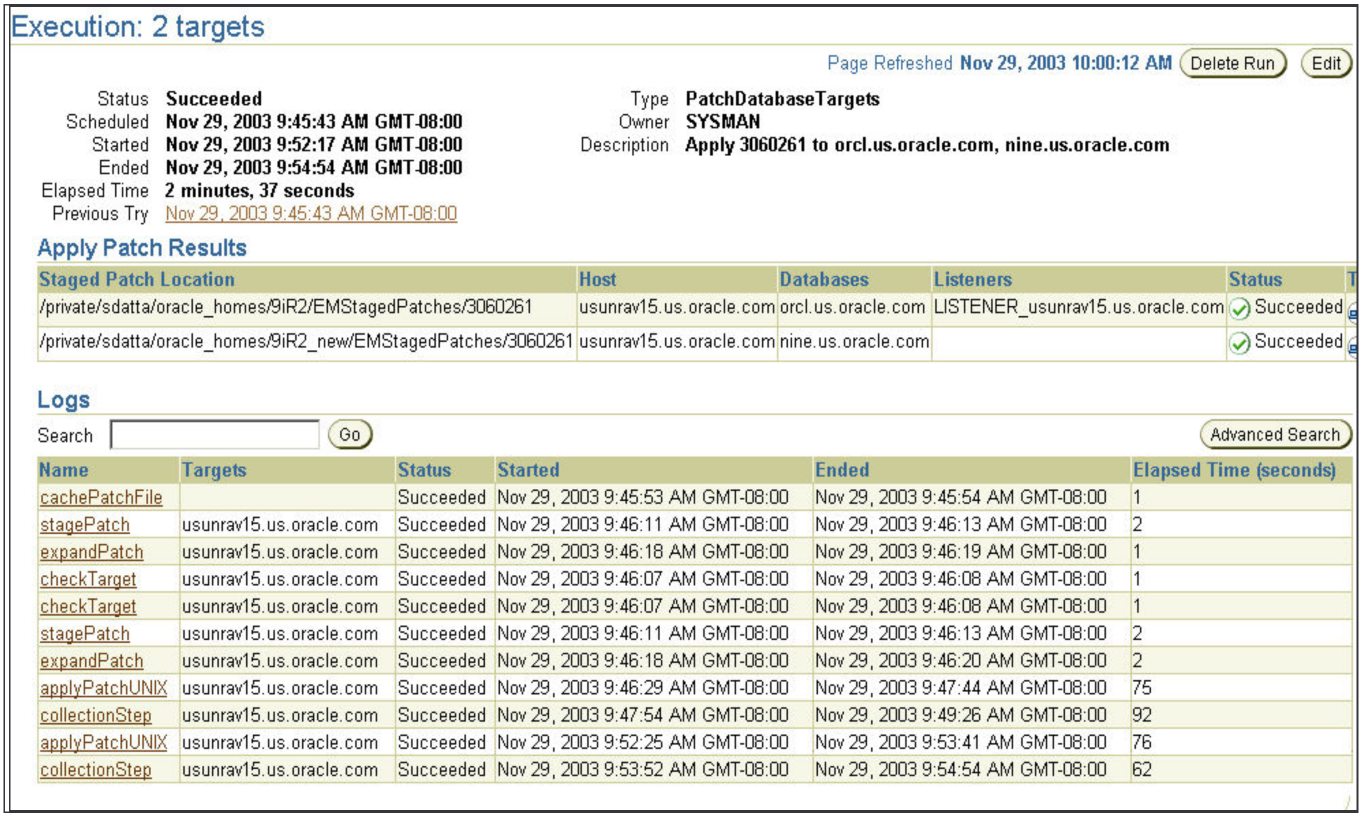
Job

Job Type	**Patch**	Target Type	**Database**
Scheduled	**Nov 29, 2003 9:25:00 AM**	Platform	**Solaris Operating System (SPARC 32-bit)**
Job Name	**Patch Job 11/29/03 9:26 AM**	Release	**9.2.0.4**
Description	**Apply 3060261 to orcl.us.oracle.com, nine.us.oracle.com**		

If the patch application fails for some reason, the status of the failed step will shows up as “Failed”. If one drills down on the failed step it will display a detailed log. In this particular example, the oracle executable is “active” and hence pre-requisite check for patching fails.

```
OPatch Version 1.0.0.0.46
Perl Version □□□

Oracle Home = /private/sdatta/oracle_homes/9iR2
Location of Oracle Inventory = /private/sdatta/oracle_homes/9iR2/inventory
Oracle Universal Installer shared library = /private/sdatta/oracle_homes/9iR2/oui/bin/solaris/liboraInstaller.so
Path to Java = /private/sdatta/oracle_homes/9iR2/jre/1.3.1/bin/java
Location of Oracle Inventory Pointer = /var/opt/oracle/oraInst.loc
Location of Oracle Universal Installer components = /private/sdatta/oracle_homes/9iR2/oui
Required Jar File under Oracle Universal Installer = lib/OraInstaller.jar

/private/sdatta/oracle_homes/9iR2/OPatch/opatch.pl version: 1.0.0.0.46
Copyright (c) 2001,2002,2003 Oracle Corporation. All Rights Reserved.

Problems when checking for files that are active.
The following files have active processes when there should be no activity:
1.  /private/sdatta/oracle_homes/9iR2/bin/oracle



ERROR: OPatch failed during pre-reqs check.
Mon Nov 24 17:11:05 2003 - Patching failed.
```

After taking corrective action one can resume the patch application by clicking on the “Retry” button. The following screenshot shows a successful patch application for the two ORACLE_HOMEs chosen in section 4.1.2.



Execution: 2 targets

Page Refreshed **Nov 29, 2003 10:00:12 AM**

Status	**Succeeded**	Type	**PatchDatabaseTargets**
Scheduled	**Nov 29, 2003 9:45:43 AM GMT-08:00**	Owner	**SYSMAN**
Started	**Nov 29, 2003 9:52:17 AM GMT-08:00**	Description	**Apply 3060261 to orcl.us.oracle.com, nine.us.oracle.com**
Ended	**Nov 29, 2003 9:54:54 AM GMT-08:00**		
Elapsed Time	**2 minutes, 37 seconds**		
Previous Try	Nov 29, 2003 9:45:43 AM GMT-08:00		

Apply Patch Results

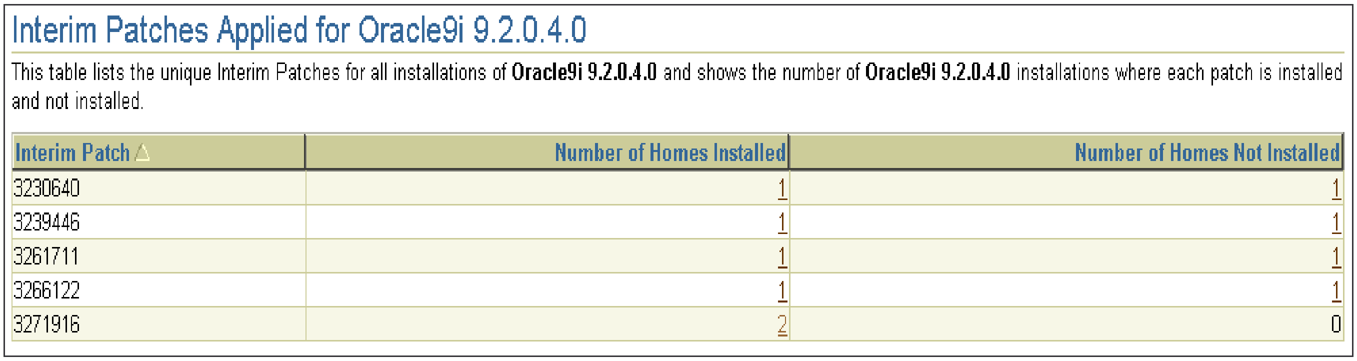
Staged Patch Location	Host	Databases	Listeners	Status	T
/private/sdatta/oracle_homes/9iR2/EMStagedPatches/3060261	usunrav15.us.oracle.com	orcl.us.oracle.com	LISTENER_usunrav15.us.oracle.com	✓ Succeeded	
/private/sdatta/oracle_homes/9iR2_new/EMStagedPatches/3060261	usunrav15.us.oracle.com	nine.us.oracle.com		✓ Succeeded	

Logs

Search

Name	Targets	Status	Started	Ended	Elapsed Time (seconds)
cachePatchFile		Succeeded	Nov 29, 2003 9:45:53 AM GMT-08:00	Nov 29, 2003 9:45:54 AM GMT-08:00	1
stagePatch	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:11 AM GMT-08:00	Nov 29, 2003 9:46:13 AM GMT-08:00	2
expandPatch	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:18 AM GMT-08:00	Nov 29, 2003 9:46:19 AM GMT-08:00	1
checkTarget	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:07 AM GMT-08:00	Nov 29, 2003 9:46:08 AM GMT-08:00	1
checkTarget	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:07 AM GMT-08:00	Nov 29, 2003 9:46:08 AM GMT-08:00	1
stagePatch	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:11 AM GMT-08:00	Nov 29, 2003 9:46:13 AM GMT-08:00	2
expandPatch	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:18 AM GMT-08:00	Nov 29, 2003 9:46:20 AM GMT-08:00	2
applyPatchUNIX	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:46:29 AM GMT-08:00	Nov 29, 2003 9:47:44 AM GMT-08:00	75
collectionStep	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:47:54 AM GMT-08:00	Nov 29, 2003 9:49:26 AM GMT-08:00	92
applyPatchUNIX	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:52:25 AM GMT-08:00	Nov 29, 2003 9:53:41 AM GMT-08:00	76
collectionStep	usunrav15.us.oracle.com	Succeeded	Nov 29, 2003 9:53:52 AM GMT-08:00	Nov 29, 2003 9:54:54 AM GMT-08:00	62

Once the interim patch has been successfully applied, it is visible by clicking on "Interim patches" for the associated targets in the Home page.



Interim Patches Applied for Oracle9i 9.2.0.4.0

This table lists the unique Interim Patches for all installations of **Oracle9i 9.2.0.4.0** and shows the number of **Oracle9i 9.2.0.4.0** installations where each patch is installed and not installed.

Interim Patch △	Number of Homes Installed	Number of Homes Not Installed
3230640	1	1
3239446	1	1
3261711	1	1
3266122	1	1
3271916	2	0

4.2 THE BACKGROUND OPERATIONS

The back end processing consists mainly of Management Server processing, Agent Job subsystem and the patch engine on the target ORACLE_HOME. Once the job is created, the Management server passes it to the agent on the target box, which then invokes the patch engine at the scheduled time.

There are two types of patch jobs for the database target: "StageDatabasePatchTargets" – a job that just downloads a patch from MetaLink and stages it to the customer's remote host; and "PatchDatabaseTargets" -a job that not only stages the patch, but also applies the patch at the remote host.

The patch jobs are so designed that:

- They are able to patch one or more database targets in parallel with a single patch per job.
- They are able to patch database targets that are multiple instances in the same ORACLE_HOME as well as part of a RAC cluster.
- Customer will have to take care of service shutdowns and startups based on the nature of the patch. Enterprise manager facilitates this by exposing a modifiable script in the Patch Wizard.
- Upon successful completion of applying the patch, the EM repository must be refreshed with the contents of the latest OUI inventory from the target host.
- All remote operations to target hosts are done through the Oracle Management Agent.
- No database that contains the EM repository and needs to be shutdown can be patched by the patch job.

The Patch job is implemented using the EM Job system. The Patch job is described in XML as a set of parameters and a series of steps that will use the parameters to execute the commands needed to patch the target or targets. The inputs provided in the Patch wizard translate into parameters for the job.

The parameters are:

- patch_id - a parameter containing the ARU ids for the patches selected.
- patch_type - a parameter containing the type of patch (either Patch or Patchset).
- apply_script - a parameter containing the lines of the script entered as the apply script.
- ARU_URL - a parameter with the value of the MetaLink URL to query for downloading patches.
- target_list – this consists of multiple entries each containing:
    - targetName = a parameter containing the database target name
    - targetType = a parameter containing the target type - “oracle_database” for example.

4.3 ORDER OF OPERATIONS

4.3.1 CACHEPATCHFILE - DOWNLOADING PATCHES INTO THE OMS REPOSITORY PATCH CACHE

The first step downloads the patch for the patch job in the ECM repository table GMT_ECM_PATCH_CACHE. This is accomplished by making a call to the EM job command “cachePatchFile” for the patch specified in the patch_id parameter passing the ARU_ID and ARU_URL as parameters:

If the patch fails to be loaded into the MGMT_ECM_PATCH_CACHE, the job will abort with an error.

4.3.2 CHECKTARGET - ESTIMATING DISK SPACE REQUIREMENTS ON THE TARGET SYSTEM

After the patch has been verified as being present in the cache, the size of the patch in bytes obtained from MetaLink is used to determine the availability of sufficient disk space on the target host. The algorithm to determine how much space is needed will be calculated as three times the size in bytes of the patch plus the size of the Oracle image located in the ORACLE_HOME/bin directory.

If there is not enough disk space available for the patch, the job will terminate patching the target and continue with patching the next target if any.

4.3.3 STAGEPATCH – COPY THE PATCH FROM OMS REPOSITORY TO TARGET HOST

If there appears to be sufficient space on the target host to stage the patch, it is copied to the remote host. This step copies the patch zip file from the OMS repository to a sub directory created under the Oracle home of the target named EMStagedPatches/<patch_id>, where patch_id is the number of the patch being staged.

4.3.4 EXPANDPATCH – UNZIPPING THE STAGED PATCH IN THE ORACLE HOME

Once the patch file has been copied to the remote host, it is unzipped at the remote host to prepare it for applying the patch. If the customer only requested that the patch be staged, the job is complete and the next steps will not appear in the job log and the job will be complete. Otherwise, the patch will be applied as described in the next two steps.

After the patch file has been copied to the target host, it will be installed. Depending on whether the host is a UNIX host or a Windows host, one of the following two steps will be run.

4.3.5 APPLYPATCHUNIX OR APPLYPATCHWIN - INSTALLING THE PATCH ON UNIX OR WINDOWS

In this step the patch is applied on the remote host by executing the script provided in the fourth step of the Patch Wizard.

4.3.6 COLLECTIONSTEP - ECM HOST INVENTORY REFRESH OF PATCH DATA

When patch has been successfully applied to the target host, the host inventory data in the OMS repository will automatically be refreshed with the latest patch information. This will be accomplished by a call to the "ConfigurationCollection" job.

4.4 PATCHING ADDITIONAL DATABASE TARGETS

The patch job iterates over the list of database targets supplied as the target_list parameter executing the steps mentioned in section 4.3 in parallel. The CachePatchFile is however done only once per patch irrespective of the number of targets.

5 THE ORACLE SOFTWARE INVENTORY

At the core of the Configuration Pack is the Repository. The repository gets the information about software installed in individual hosts from the OUI (Oracle Universal Installer) inventories. The inventories are satellites of software related information for each ORACLE_HOME. Within each host there are two types of OUI inventories:

The Local Inventory: The local inventory resides inside the inventory subdirectory within each ORACLE_HOME. The inventory consists of a file called comps.xml, which contains all the components as well as patchsets or interim patches installed in the ORACLE_HOME. It also contains details about non-Oracle components like Java Runtime Environment (JRE) required by different Java based Oracle tools and components. With OUI 2.1 and higher, the information in inventory is stored in Extensible Markup Language (XML) format. The XML format allows for easier diagnosis of problems and faster loading of data. Any secure information is not stored directly in the inventory. As a result, during deinstallation of some products, one may be prompted for required secure information, such as a password.

The Central Inventory: The central inventory is an inventory that lists a set of ORACLE_HOMES installed in the host. Each central inventory consists of a file called inventory.xml, which contains the list. Following is an extract from the inventory.xml file for a host that has RDBMS 9i, RDBMS 10g and Enterprise Manager grid Control installed.

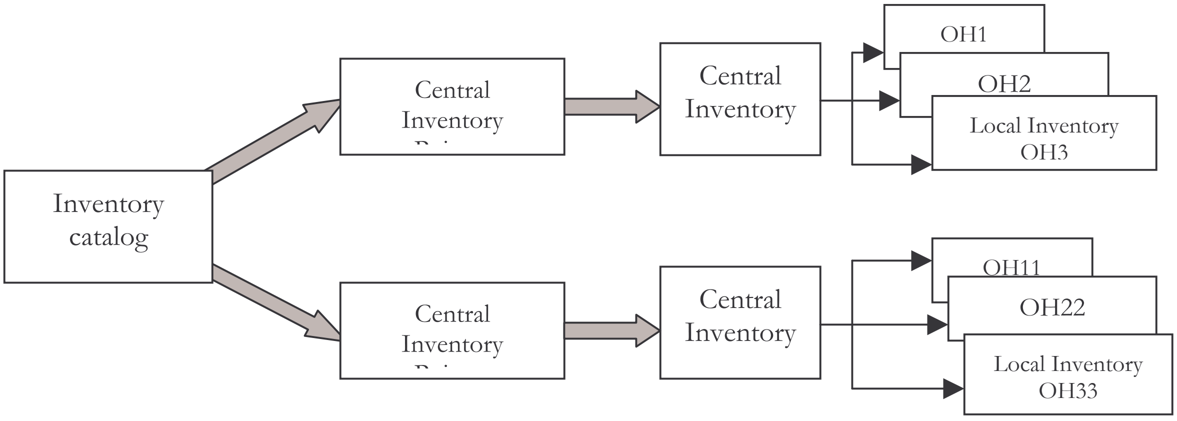
```
<?xml version="1.0" standalone="yes" ?>
<!-- Copyright (c) 2002 Oracle Corporation. All rights Reserved -->
<!-- Do not modify the contents of this file by hand. -->
```

```
<INVENTORY>
<VERSION_INFO>
  <SAVED_WITH>10.1.0.2.0</SAVED_WITH>
  <MINIMUM_VER>2.1.0.6.0</MINIMUM_VER>
</VERSION_INFO>
<HOME_LIST>
<HOME NAME="DB_NINE" LOC="/net/usunrav15/private/sdatta/oracle_homes/9iR2" TYPE="O" IDX="1"/>
<HOME NAME="EM_HOME" LOC="/net/usunrav15/private/sdatta/oracle_homes/EM" TYPE="O" IDX="2"/>
<HOME NAME="TEN_HOME" LOC="/private/sdatta/oracle_homes/10g" TYPE="O" IDX="3"/>
</HOME_LIST>
</INVENTORY>
```

Hosted environments, typically found in an Application Service Provider (ASP), however, pose a challenge for a single Central inventory model. In a hosted model, each host houses applications for different organizations. Apart from obvious security problems arising out of a "well published" central inventory, each of these organizations may have different operational practices and demand exclusive access to the central inventory for conducting lifecycle operations like patching or upgrading. This demands a mutually segregated number of central inventories each pointed to by an inventory pointer file.

For default installations the inventory pointer file is located in a platform specific "well-known" location e.g. /var/opt/oracle/oraInst.loc on Solaris. On Windows the pointer is in the registry. Each "set of installations" will have its own central inventory pointer file, unknown to other "set of installations". Operations like patching, upgrading and installing support command line arguments for explicitly specifying the central inventory pointer location. Only the agent is aware of all the central inventory locations since it maintains a catalog of all central inventory pointer locations in a file. Hosts, which are not shared by groups, can however, operate with a single central inventory.

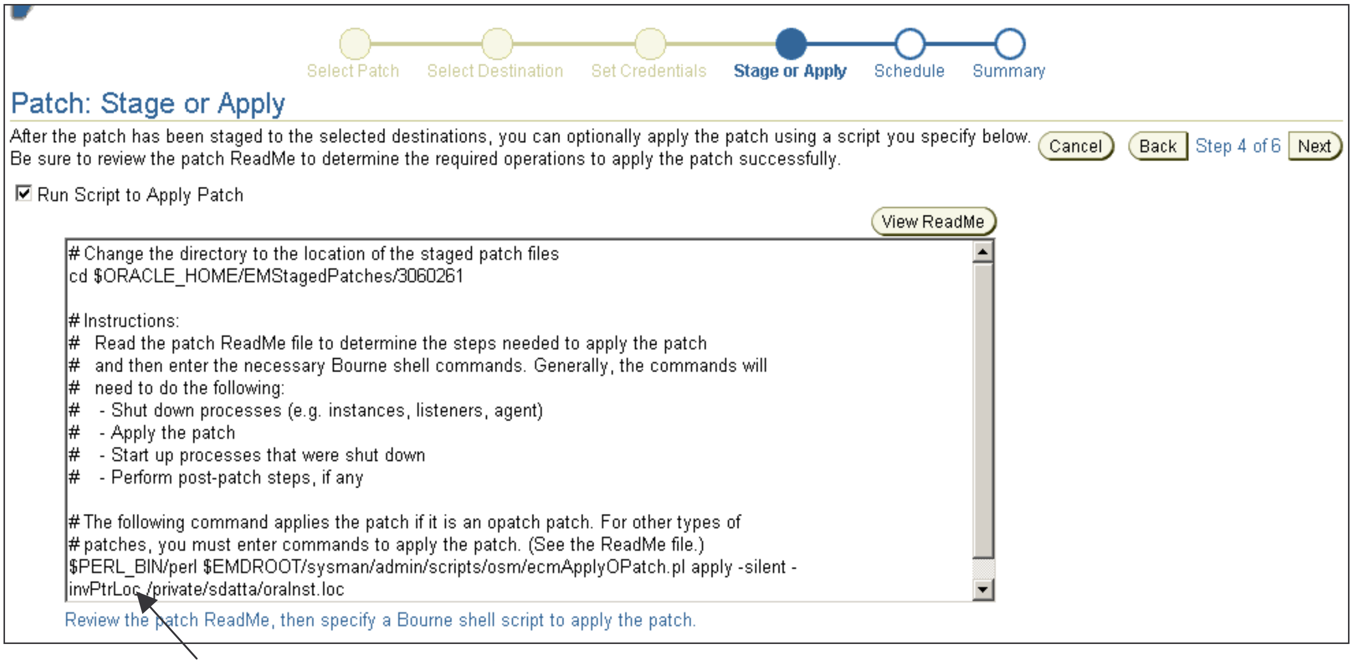
All read and write operations on inventories are performed by the Oracle Universal Installer components. During patching these components are invoked by the patch engine through APIs so that appropriate compatibility and conflict checks are performed and finally the inventory is updated with the interim patch information once the patch has been successfully applied. Currently, operations on a Central inventory are serialized through a locking mechanism. This implies that while an operation like install, upgrade or patch happens on an ORACLE_HOME, such operations will get blocked on other ORACLE_HOMEs that share the same Central inventory. For patching however, the life of such locks are very short, patching operations on other ORACLE_HOMEs would try to access the inventory through a simple wait and retry mechanism.



6 Handling specific cases

6.1 Patching an ORACLE_HOME with non-default Central inventory pointer location

The Inventory pointer location has to be passed as an argument to the patch tool. In the example below, the Central inventory is pointed to by the contents of /private/sdatta/oraInst.loc.



Patch: Stage or Apply

After the patch has been staged to the selected destinations, you can optionally apply the patch using a script you specify below. Be sure to review the patch ReadMe to determine the required operations to apply the patch successfully.

☑ Run Script to Apply Patch

```
# Change the directory to the location of the staged patch files
cd $ORACLE_HOME/EMStagedPatches/3060261

# Instructions:
#   Read the patch ReadMe file to determine the steps needed to apply the patch
#   and then enter the necessary Bourne shell commands. Generally, the commands will
#   need to do the following:
#    - Shut down processes (e.g. instances, listeners, agent)
#    - Apply the patch
#    - Start up processes that were shut down
#    - Perform post-patch steps, if any

# The following command applies the patch if it is an opatch patch. For other types of
# patches, you must enter commands to apply the patch. (See the ReadMe file.)
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent -
invPtrLoc /private/sdatta/oraInst.loc
```

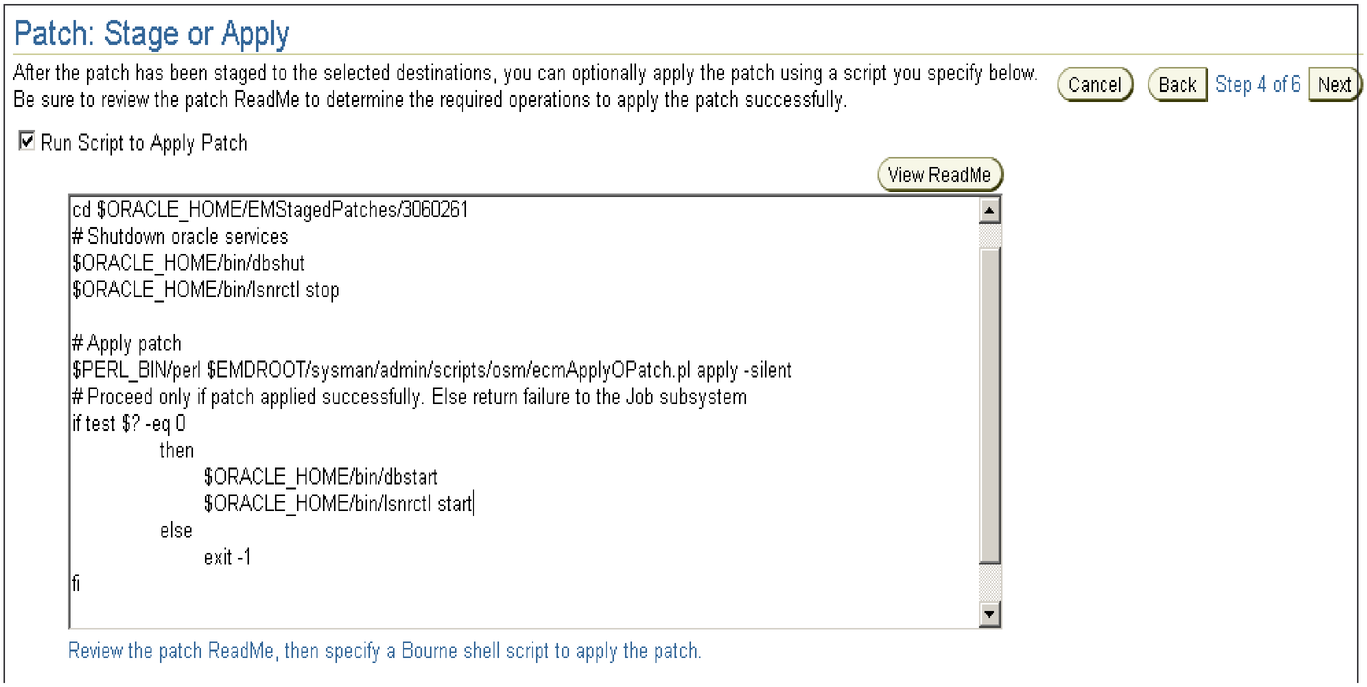
Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

Inventory Pointer Location

6.2 Automating shutdown and startup of Oracle services

For 10*g* patches, the directives for pre and post patch processing will be handled from the patch itself. For 8i and 9i patches, however, the patch wizard is not aware of handling service level requirements for patching. However, it is possible to automate the sequence of instructions in stage 4 of the Patch Wizard. Following is an example where the README states that instances and the listener need to be shutdown before patching and need to be brought up after patching. The instance can be shutdown in two ways, either by writing custom scripts or by using the "dbshut" script provided out of box with the database (the out of box "dbshut script" executes "shutdown" rather than "shutdown immediate". One may want to modify this to "shutdown immediate" so that it does not wait on existing connections). For "dbshut" to work on all instances of a home the third field in the oratab file has to be set to "Y" for all the instances belonging to the ORACLE_HOME. In this example, the ORACLE_HOME to be patched is "/private/sdatta/oracle_homes/9iR2".

Entry in /var/opt/oracle/oratab: *:/private/sdatta/oracle_homes/9iR2:**Y**



Patch: Stage or Apply

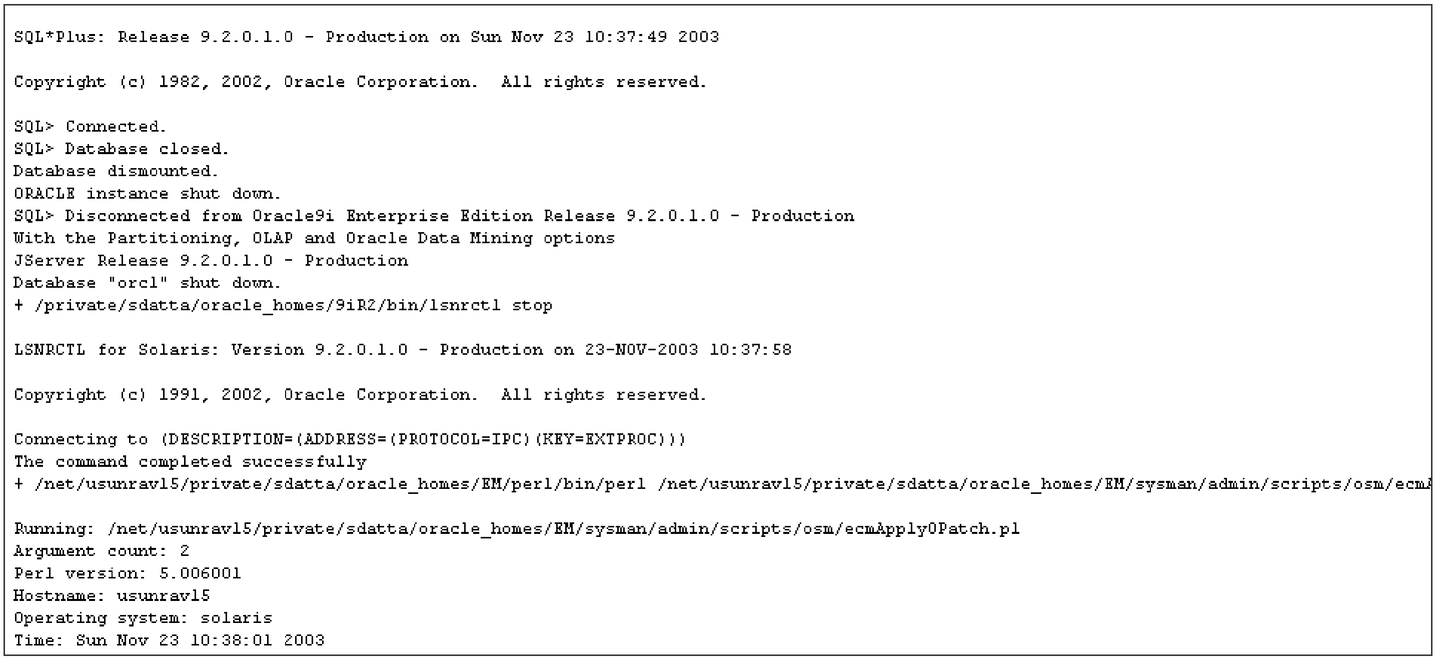
After the patch has been staged to the selected destinations, you can optionally apply the patch using a script you specify below. Be sure to review the patch ReadMe to determine the required operations to apply the patch successfully.

☑ Run Script to Apply Patch

```
cd $ORACLE_HOME/EMStagedPatches/3060261
# Shutdown oracle services
$ORACLE_HOME/bin/dbshut
$ORACLE_HOME/bin/lsnrctl stop

# Apply patch
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent
# Proceed only if patch applied successfully. Else return failure to the Job subsystem
if test $? -eq 0
        then
            $ORACLE_HOME/bin/dbstart
            $ORACLE_HOME/bin/lsnrctl start
        else
            exit -1
fi
```

Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

Once the job has been submitted, one can view the Job log to see if the above steps are executed successfully. The following screenshot shows that the listener and the database are successfully stopped before the patch engine has been invoked. If the steps are successfully executed, one may want to include these as a part of a reusable script called say, “shutdown_patch_startup.sh” and keep it in the agent home so that it can be reused over and over again for patches that demand such functionality.



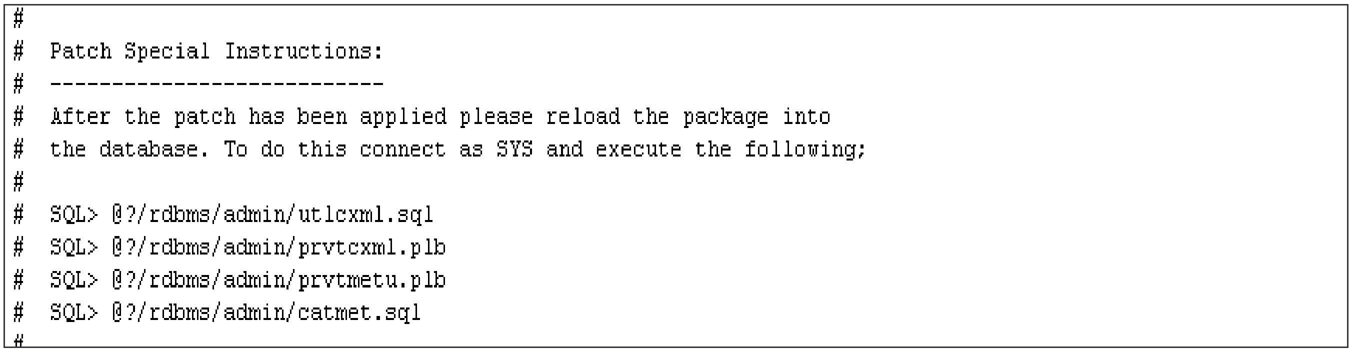
```
SQL*Plus: Release 9.2.0.1.0 - Production on Sun Nov 23 10:37:49 2003

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
Database "orcl" shut down.
+ /private/sdatta/oracle_homes/9iR2/bin/lsnrctl stop

LSNRCTL for Solaris: Version 9.2.0.1.0 - Production on 23-NOV-2003 10:37:58

Copyright (c) 1991, 2002, Oracle Corporation.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
The command completed successfully
+ /net/usunravl5/private/sdatta/oracle_homes/EM/perl/bin/perl /net/usunravl5/private/sdatta/oracle_homes/EM/sysman/admin/scripts/osm/ecmA

Running: /net/usunravl5/private/sdatta/oracle_homes/EM/sysman/admin/scripts/osm/ecmApplyOPatch.pl
Argument count: 2
Perl version: 5.006001
Hostname: usunravl5
Operating system: solaris
Time: Sun Nov 23 10:38:01 2003
```

6.3 APPLYING A SQL PATCH

Occasionally, Oracle releases patches that consist of SQL and PLSQL scripts. This is common to patches pertaining to certain components like Intermedia and Streams where database objects form integral elements of the software.

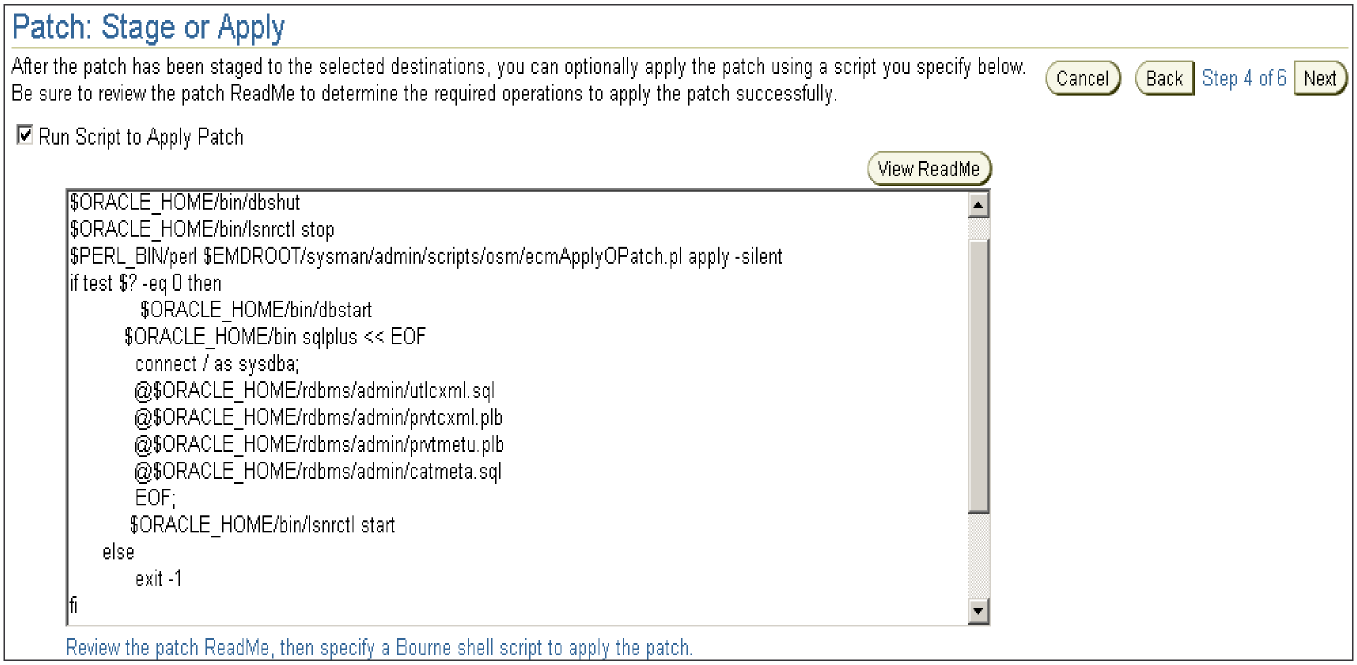
Technically, users can apply the patch on a fully running database. But this is not recommended. For avoiding any inconsistent behavior it is recommended to bounce the database in restricted mode and then apply the SQL changes to the database. In this example, we have chosen to startup the database in normal mode but kept the listener down so that it is inaccessible to other users. The listener is brought up once the patch application is successful.

For 10g patches we expect patch specific SQL directives to be provided along with the patch and to be handled by the post patch script handling capability of opatch. However, this is not a reality for already published 8i and 9i patches. For those patches we need to script special directives so that the job subsystem can execute them as post patching action. Let us pick patch 3193124 for Solaris. This applies on top of 9.2.0.4 and consists of four database scripts as shown in the extract from the patch README:



```
#
#  Patch Special Instructions:
#  ---------------------------
#  After the patch has been applied please reload the package into
#  the database. To do this connect as SYS and execute the following;
#
#  SQL> @?/rdbms/admin/utlcxml.sql
#  SQL> @?/rdbms/admin/prvtcxml.plb
#  SQL> @?/rdbms/admin/prvtmetu.plb
#  SQL> @?/rdbms/admin/catmet.sql
#
```

In this particular example, we have passed commands to shutdown the Oracle services before applying the patch. After the patch has been successfully applied, we will start the database but not the listener to prevent connections from clients. Alternatively we can startup in restricted mode. Once the database is started we will apply the SQL changes one by one. We need to be careful about return code handling since the job subsystem processes the return code of the last statement. However, we want the whole job to fail in case the ‘opatch apply’ is unsuccessful. Therefore, we trap the return code of the opatch command and process the return code conditionally within a conditional loop.



Patch: Stage or Apply

After the patch has been staged to the selected destinations, you can optionally apply the patch using a script you specify below.
Be sure to review the patch ReadMe to determine the required operations to apply the patch successfully.

☑ Run Script to Apply Patch

```
$ORACLE_HOME/bin/dbshut
$ORACLE_HOME/bin/lsnrctl stop
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent
if test $? -eq 0 then
            $ORACLE_HOME/bin/dbstart
         $ORACLE_HOME/bin sqlplus << EOF
           connect / as sysdba;
           @$ORACLE_HOME/rdbms/admin/utlcxml.sql
           @$ORACLE_HOME/rdbms/admin/prvtcxml.plb
           @$ORACLE_HOME/rdbms/admin/prvtmetu.plb
           @$ORACLE_HOME/rdbms/admin/catmeta.sql
           EOF;
          $ORACLE_HOME/bin/lsnrctl start
     else
           exit -1
fi
```

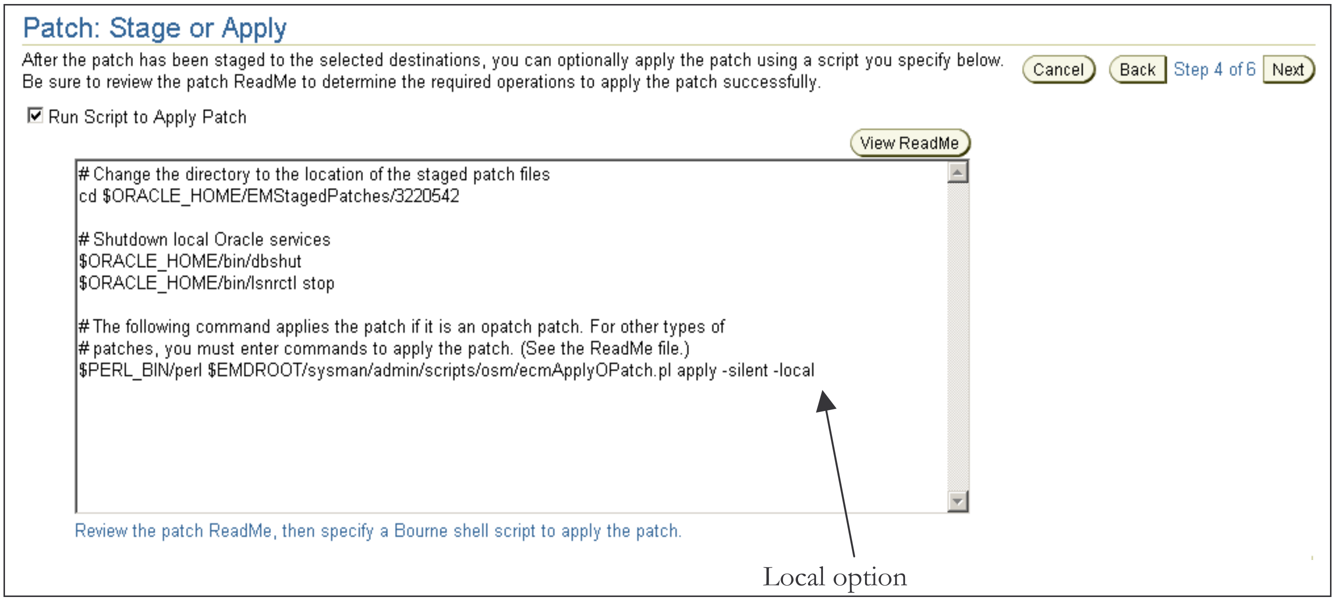
Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

6.4 Applying a RAC patch

Patching a RAC environment is slightly more complex than patching a single node. With the exception of rolling patch case (see the next section for rolling patches), the RAC nodes should not operate with different levels of Oracle software. We can achieve this in two different ways:

- Shutdown all instances and patch all instances (this will cause a bigger downtime).
- Shutdown one RAC instance at a time and make sure that the instance is not restarted unless all the nodes are patched. This will cause minimal downtime. Since we cannot possibly synchronize the shutdown and startup from Enterprise Manager, we will only shutdown every instance prior to patching.

It is recommended that we patch one node at a time and make sure we do not propagate changes to other nodes. This is important because we do not initiate automatic collection on nodes where the patch is propagated automatically. This may cause inconsistency between Enterprise Manager and the actual inventory.



Patch: Stage or Apply

After the patch has been staged to the selected destinations, you can optionally apply the patch using a script you specify below. Be sure to review the patch ReadMe to determine the required operations to apply the patch successfully.

☑ Run Script to Apply Patch

```
# Change the directory to the location of the staged patch files
cd $ORACLE_HOME/EMStagedPatches/3220542

# Shutdown local Oracle services
$ORACLE_HOME/bin/dbshut
$ORACLE_HOME/bin/lsnrctl stop

# The following command applies the patch if it is an opatch patch. For other types of
# patches, you must enter commands to apply the patch. (See the ReadMe file.)
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent -local
```

Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

Local option

Ideally, all the instances for the RAC database should be selected in the targets selection page. In that case, this script can be rolled out to all of them so that the patching can be parallel across the RAC. The shutdown of Oracle services ensures that no two nodes can run with incompatible software patch levels.

6.5 Applying a Rolling RAC patch

6.5.1 Introduction to Rolling RAC patch

Starting Database version 9.2.0.4.0 and "opatch" version 1.0.0.46, Oracle supports the application of patches to the nodes of a Real Application Clusters (RAC) system in a rolling fashion. For applying a normal RAC patch on a three node RAC, the sequence of patching is:

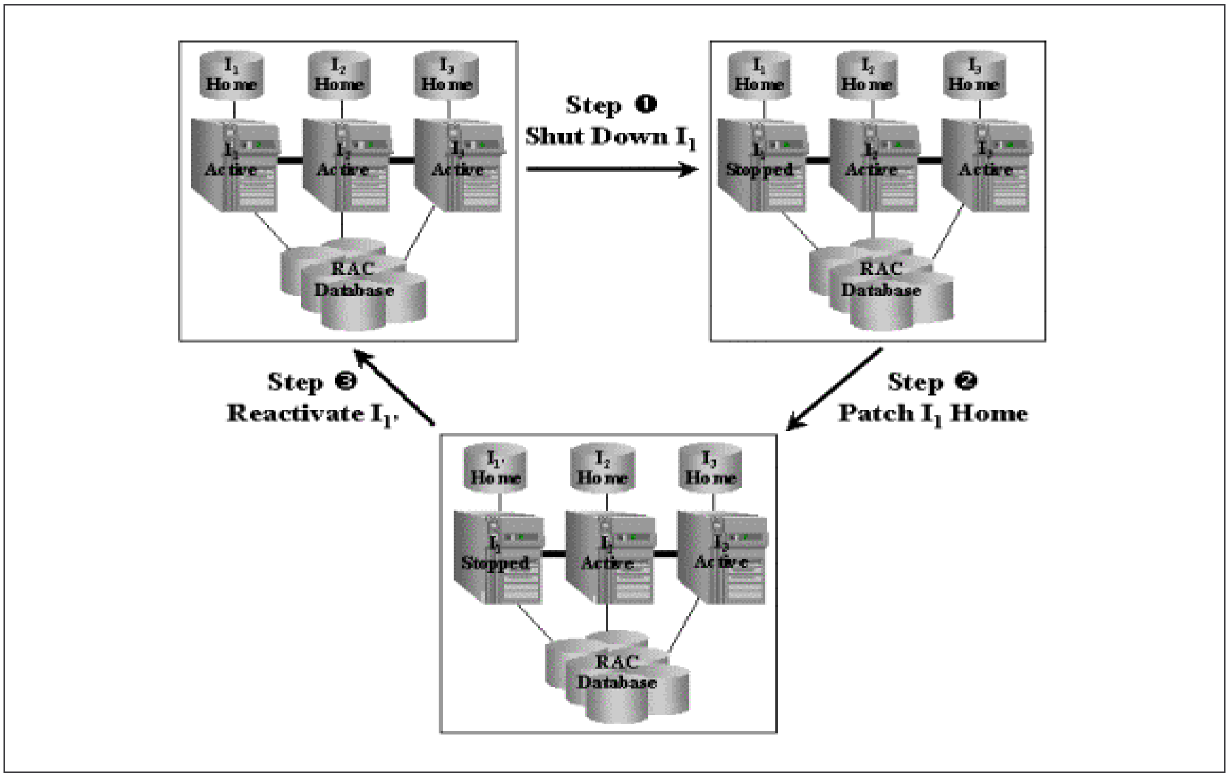
1. Shutdown Oracle instance on node 1
2. Apply patch on node 1
3. Shutdown Oracle instance on node 2
4. Apply patch on node 2
5. Shutdown Oracle instance on node 3
6. Start instance on node 1
7. Apply patch on node 3

Technically speaking, we have a downtime for the whole RAC system between steps 5 and 6.

With rolling RAC patching, the sequence of patching a three-node cluster is:

1. Shutdown Oracle instance on node 1
2. Apply patch on node 1
3. Start instance on node 1
4. Shutdown Oracle instance on node 2
5. Apply patch on node 2

6. Start instance on node 2
7. Shutdown Oracle instance on node 3
8. Apply patch on node 3
9. Start instance on node 3



A RAC system runs with all nodes actively processing transactions on the behalf of database clients (upper left hand RAC system in the above figure). Step 1 of the patch application procedure is to quiesce the first instance to which the patch is to be applied (instance 1 in this example). In step 2 an Oracle patch tool (opatch) is used to apply the patch to the quiesced instance (the Oracle Home for instance 1 is updated). In step 3 the patched instance is reactivated and rejoins the cluster. The RAC system is now running with one instance at a higher maintenance level than the other nodes in the cluster.

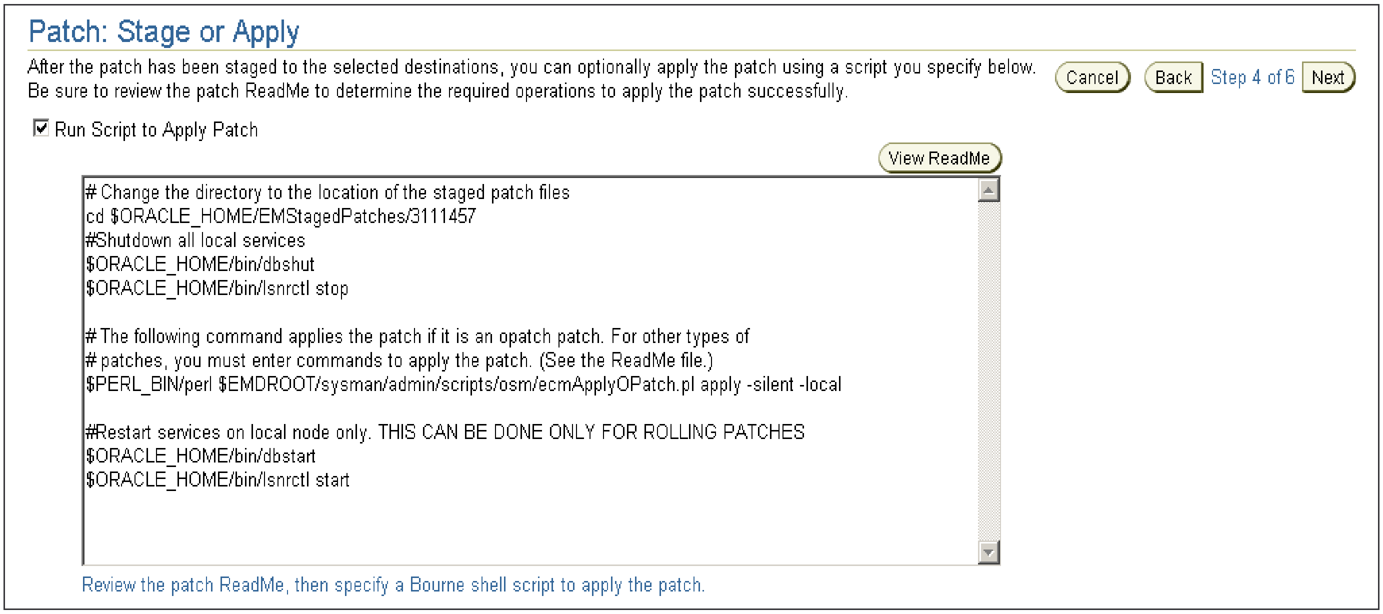
A RAC system can run in this mixed mode for an arbitrary period to test the patch in the production environment. When satisfied that the patch corrects the original problem and has not introduced a problem, the procedure is repeated for the remaining nodes in the cluster. When all nodes in the cluster have been patched (using this method) the rolling patch update is complete and all nodes are running the same version of Oracle

However, not all patches can be applied in a rolling fashion. The patches that can be considered “rollable” include, but are not limited to:

- Patches that change procedural logic and do not touch common header definitions for any kernel module. Most client side patches that only affect utilities like export, import, sql*plus, sql*loader etc fals in this category.
- Patches that do not affect the contents of the RAC database.

- Patches that are not related to the RAC internode communication infrastructure.

Oracle development while producing a fix for a bug can mark a patch as "rollable", if the criteria are met. For a rolling RAC patch, we will apply the patch in the same manner as a regular RAC patch except that we can bring the Oracle services up immediately after patching.



Patch: Stage or Apply

After the patch has been staged to the selected destinations, you can optionally apply the patch using a script you specify below. Be sure to review the patch ReadMe to determine the required operations to apply the patch successfully.

☑ Run Script to Apply Patch

```
# Change the directory to the location of the staged patch files
cd $ORACLE_HOME/EMStagedPatches/3111457
#Shutdown all local services
$ORACLE_HOME/bin/dbshut
$ORACLE_HOME/bin/lsnrctl stop

# The following command applies the patch if it is an opatch patch. For other types of
# patches, you must enter commands to apply the patch. (See the ReadMe file.)
$PERL_BIN/perl $EMDROOT/sysman/admin/scripts/osm/ecmApplyOPatch.pl apply -silent -local

#Restart services on local node only. THIS CAN BE DONE ONLY FOR ROLLING PATCHES
$ORACLE_HOME/bin/dbstart
$ORACLE_HOME/bin/lsnrctl start
```

Review the patch ReadMe, then specify a Bourne shell script to apply the patch.

6.6 Applying patch on a NFS mounted home

It is customary to run instances on different nodes from a single shared ORACLE_HOME. This ORACLE_HOME could be residing on a NetApp filer in the form of a Network File System (NFS) mounted home or on a Clustered File system (CFS). There could be several "consumers" of such a shared ORACLE_HOME- database instances, listeners etc. In such cases the ORACLE_HOME can only be patched from the node which instantiated the inventory for that ORACLE_HOME either through install or clone. It is therefore important that for patches, which require service shutdowns, that one manually shuts down the targets that are the running off that ORACLE_HOME.

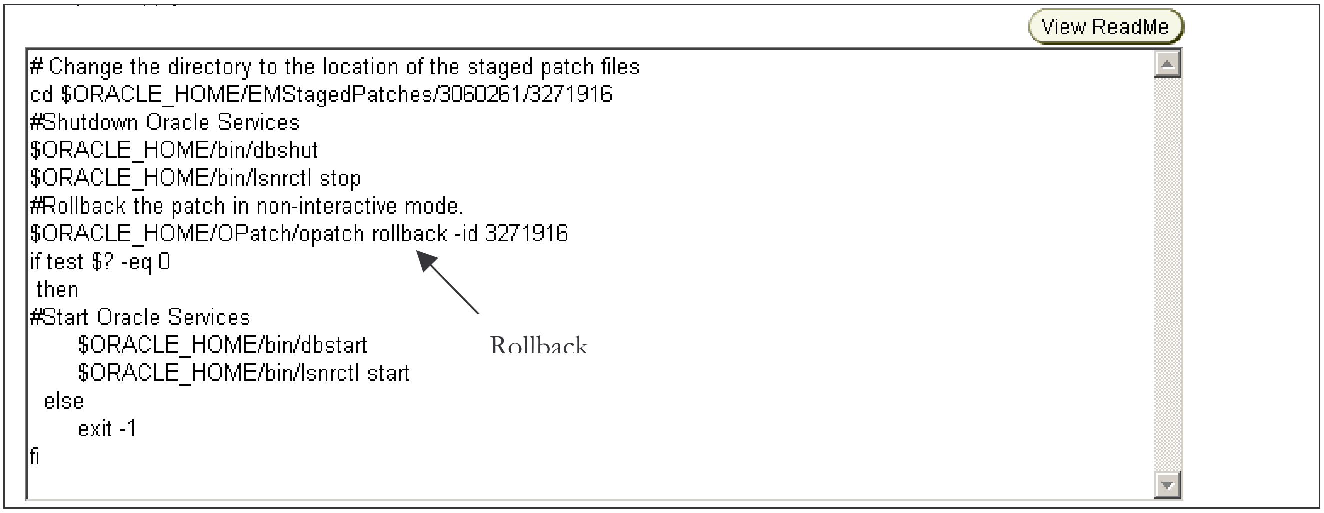
7 Rolling back patches

Rarely patches need to be rolled back for various reasons. Enterprise Manager does not provide any particular wizard based facility to rollback patches. Rollback can be done in two ways:

- It is possible to schedule an Enterprise Manager job that does the same. Roll back of patches could have similar validations as the "apply" including checks if there are active processes.
- One can use the "Stage or Apply" screen to rollback the patch. Though this could be confusing, one can use the script to rollback across several nodes.

It is interesting to note that while the application of patch starts with the selection of the base bug, the rollback is based on the platform specific patch installed for that platform. So one has to pass the patch number as an argument. One can find the patch associated with a base bug by using the "Search Oracle Products Installed in Oracle Homes" functionality in Enterprise Manager. In the following example patch 3271916 resolves base bug 3060261 on Solaris 32 bit.

The following screenshot shows an example:

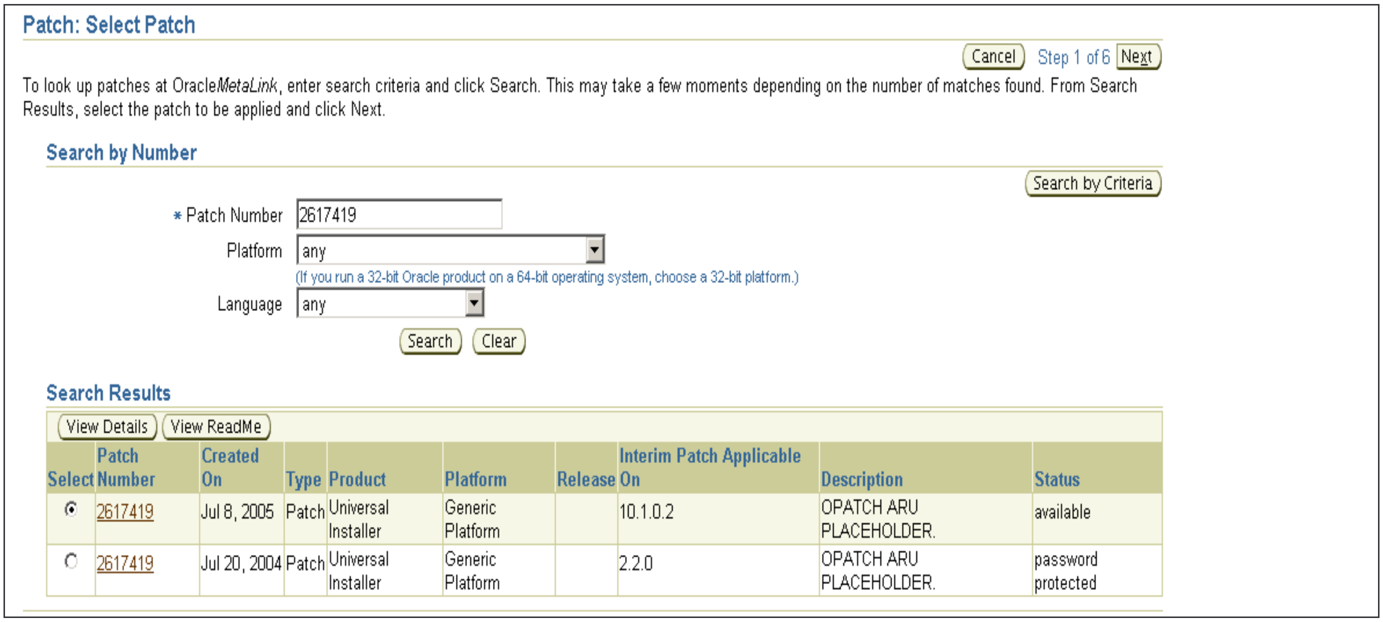


```
# Change the directory to the location of the staged patch files
cd $ORACLE_HOME/EMStagedPatches/3060261/3271916
#Shutdown Oracle Services
$ORACLE_HOME/bin/dbshut
$ORACLE_HOME/bin/lsnrctl stop
#Rollback the patch in non-interactive mode.
$ORACLE_HOME/OPatch/opatch rollback -id 3271916
if test $? -eq 0
 then
#Start Oracle Services
      $ORACLE_HOME/bin/dbstart
      $ORACLE_HOME/bin/lsnrctl start
  else
      exit -1
fi
```

8 Distributing opatch from Grid Control

Critical patch updates often recommend using the latest version of the backend patch tool, opatch.. This could be a time consuming exercise if there are many ORACLE_HOMEs and each has to be updated with the latest opatch version. To resolve this, users can use the Grid Control patching facility and the job subsystem to distribute opatch itself in a mass scalable manner. The agent on the target hosts have to be 10.1.0.4 or above to achieve this. They can do it in two ways:

- Simply search Metalink from the "Patch Search Page" with Patch Number 2617419. In the "Destination Selection" page, they have to provide the "Host and Directory" for the ORACLE_HOMEs to be updated and then follow the rest of the patching wizard. Though this means that each host and ORACLE_HOME has to be entered, the advantage of this method is that such pairs can be keyed in independent of the version of opatch. So the software can be simultaneously updated to multiple ORACLE_HOMEs with different versions. Though the opatch has is staged in Metalink as a 10.1.0.2 product, it does not matter if the user chooses this method.



Patch: Select Patch

To look up patches at Oracle*MetaLink*, enter search criteria and click Search. This may take a few moments depending on the number of matches found. From Search Results, select the patch to be applied and click Next.

Search by Number

* Patch Number: 2617419

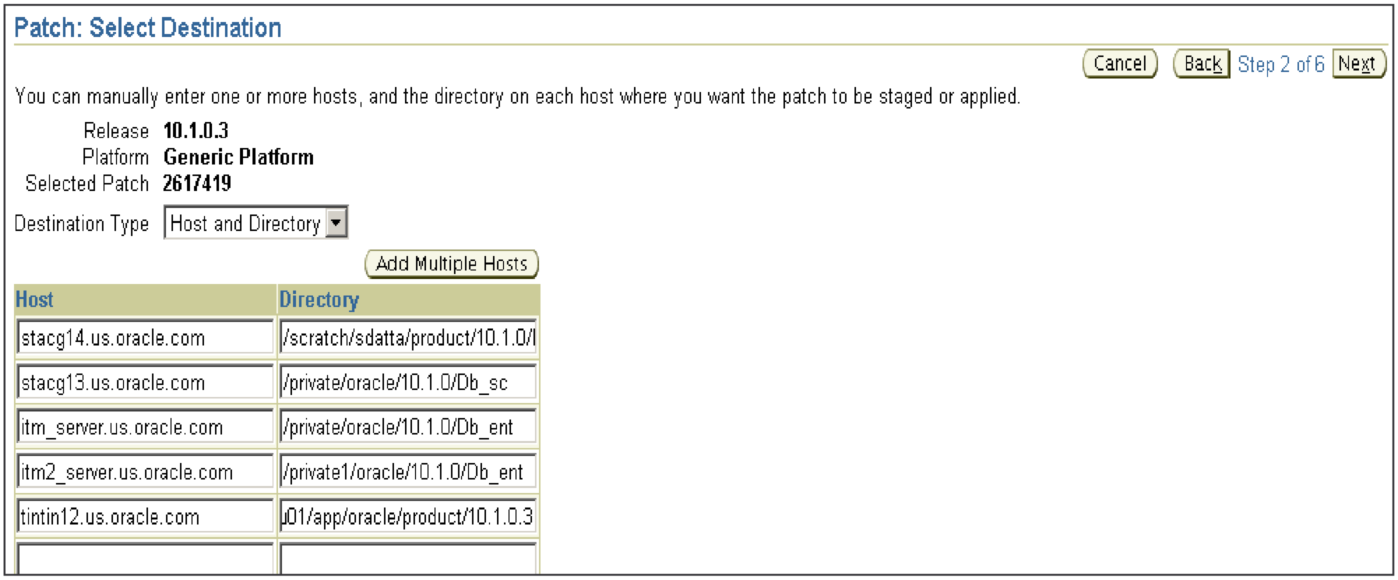
Platform: any

(If you run a 32-bit Oracle product on a 64-bit operating system, choose a 32-bit platform.)

Language: any

Search Results

Select	Patch Number	Created On	Type	Product	Platform	Release	Interim Patch Applicable On	Description	Status
●	2617419	Jul 8, 2005	Patch	Universal Installer	Generic Platform		10.1.0.2	OPATCH ARU PLACEHOLDER.	available
○	2617419	Jul 20, 2004	Patch	Universal Installer	Generic Platform		2.2.0	OPATCH ARU PLACEHOLDER.	password protected



Patch: Select Destination

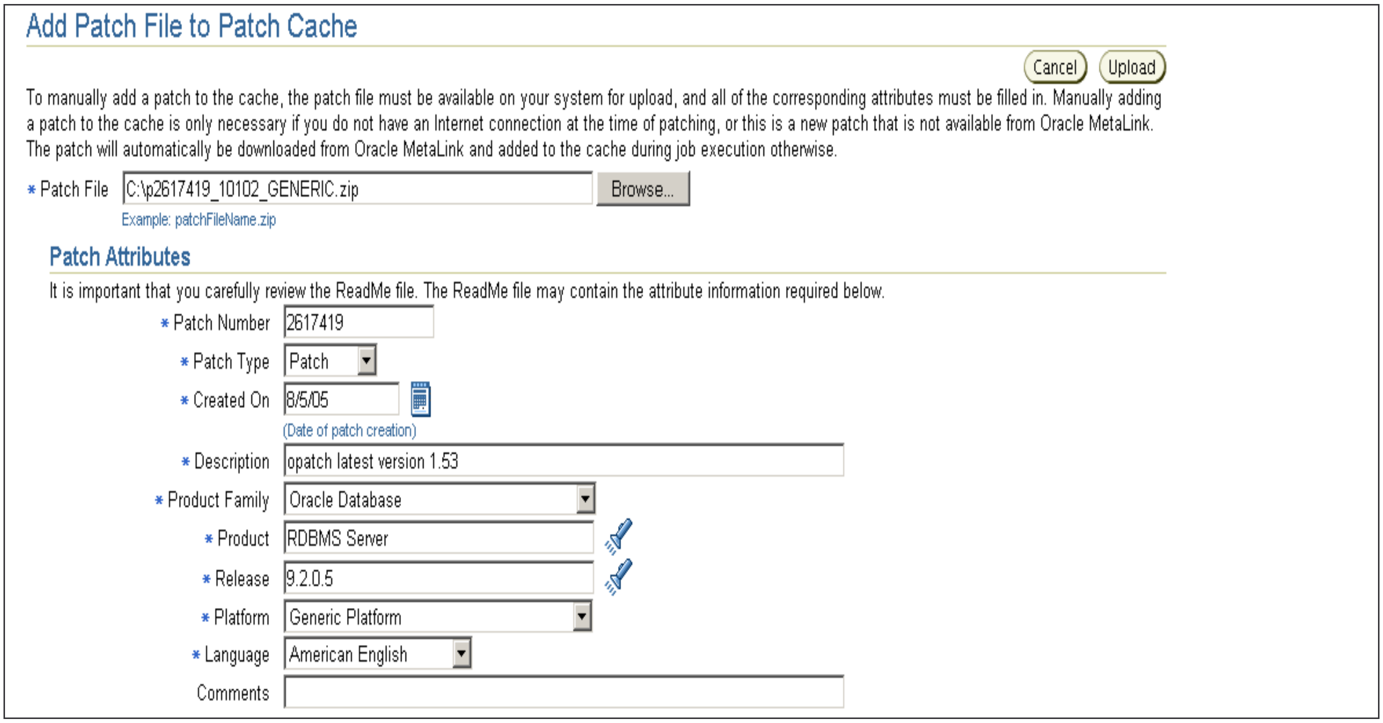
You can manually enter one or more hosts, and the directory on each host where you want the patch to be staged or applied.

Release	10.1.0.3
Platform	Generic Platform
Selected Patch	2617419
Destination Type	Host and Directory

Host	Directory
stacg14.us.oracle.com	/scratch/sdatta/product/10.1.0/
stacg13.us.oracle.com	/private/oracle/10.1.0/Db_sc
itm_server.us.oracle.com	/private/oracle/10.1.0/Db_ent
itm2_server.us.oracle.com	/private1/oracle/10.1.0/Db_ent
tintin12.us.oracle.com	u01/app/oracle/product/10.1.0.3

- If users find it convenient to distribute opatch based on specific database targets, they can also upload it as an RDBMS patch for Generic platform with the version same as the ORACLE_HOMEs they want to update and then follow the regular patching process. The relevant targets will be automatically selected. However, they have to iterate this process for each version of the target versions. The Patch Cache allows only unique patch numbers for a particular platform, so for each version of the target, the previous version of opatch has to be deleted from the Patch Cache.

  e.g. If users want to distribute opatch to a bunch of 9.2.0.5.0 RDBMS ORACLE_HOMEs, they have to upload the patch 2617419 to the "Patch Cache" as a 9.2.0.5.0 patch. The following screenshot shows this.



Add Patch File to Patch Cache

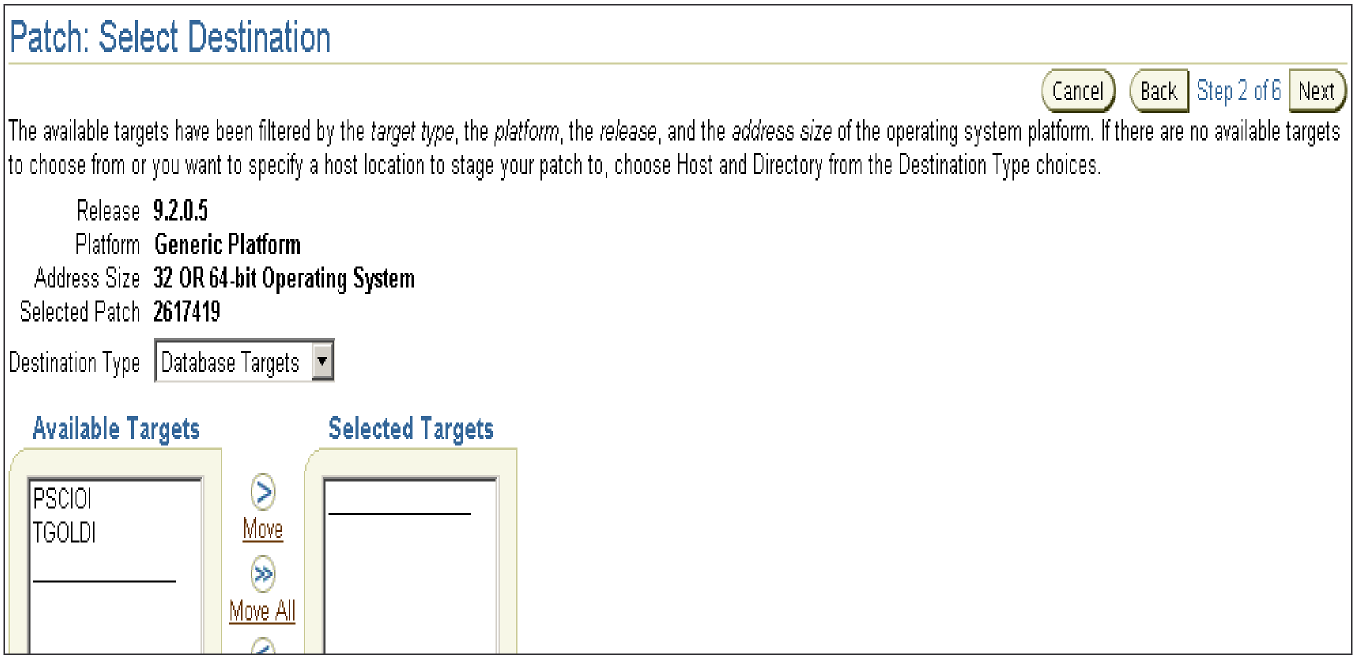
To manually add a patch to the cache, the patch file must be available on your system for upload, and all of the corresponding attributes must be filled in. Manually adding a patch to the cache is only necessary if you do not have an Internet connection at the time of patching, or this is a new patch that is not available from Oracle MetaLink. The patch will automatically be downloaded from Oracle MetaLink and added to the cache during job execution otherwise.

* Patch File: C:\p2617419_10102_GENERIC.zip
Example: patchFileName.zip

Patch Attributes

It is important that you carefully review the ReadMe file. The ReadMe file may contain the attribute information required below.

* Patch Number: 2617419
* Patch Type: Patch
* Created On: 8/5/05
(Date of patch creation)
* Description: opatch latest version 1.53
* Product Family: Oracle Database
* Product: RDBMS Server
* Release: 9.2.0.5
* Platform: Generic Platform
* Language: American English
Comments:



Patch: Select Destination

The available targets have been filtered by the *target type*, the *platform*, the *release*, and the *address size* of the operating system platform. If there are no available targets to choose from or you want to specify a host location to stage your patch to, choose Host and Directory from the Destination Type choices.

Release **9.2.0.5**
Platform **Generic Platform**
Address Size **32 OR 64-bit Operating System**
Selected Patch **2617419**

Destination Type: Database Targets

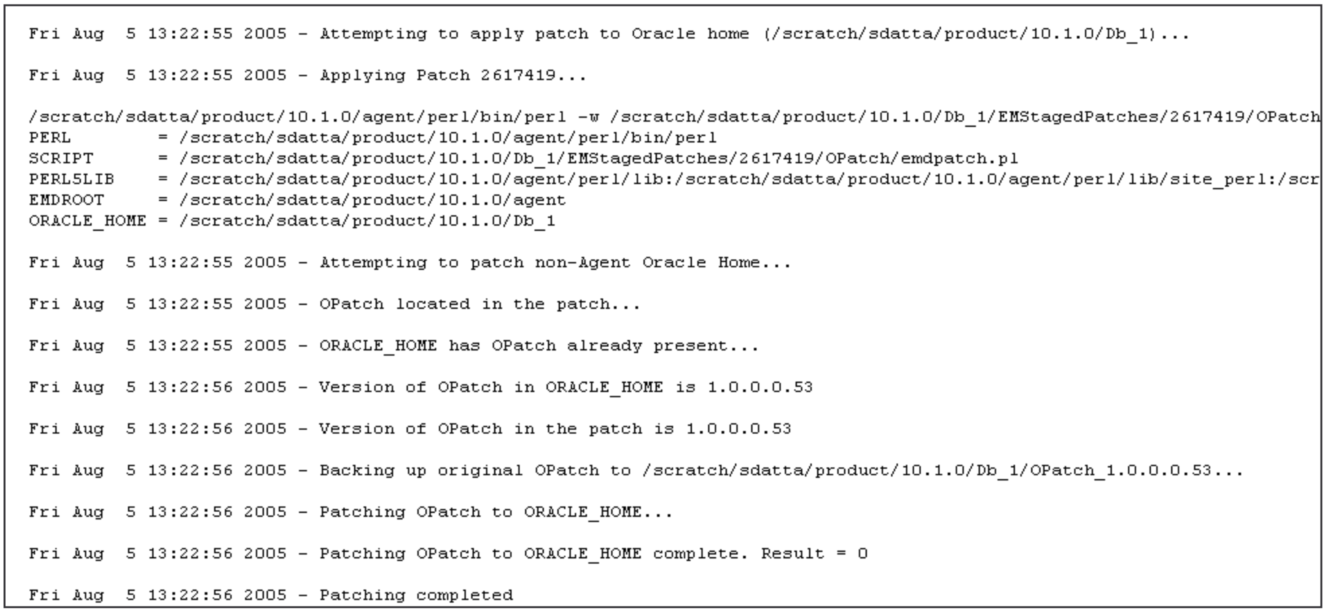
Available Targets

PSCIOI
TGOLDI

Selected Targets

Verification:

The “ApplypatchUnix” or ApplyPatchWin” step of the patching job logs the actions. One can clearly see that if an existing version of Opatch exists, the version is backed up in the ORACLE_HOME. The following output from the job log clearly shows the execution details:



```
Fri Aug  5 13:22:55 2005 - Attempting to apply patch to Oracle home (/scratch/sdatta/product/10.1.0/Db_1)...

Fri Aug  5 13:22:55 2005 - Applying Patch 2617419...

/scratch/sdatta/product/10.1.0/agent/perl/bin/perl -w /scratch/sdatta/product/10.1.0/Db_1/EMStagedPatches/2617419/OPatch
PERL        = /scratch/sdatta/product/10.1.0/agent/perl/bin/perl
SCRIPT      = /scratch/sdatta/product/10.1.0/Db_1/EMStagedPatches/2617419/OPatch/emdpatch.pl
PERL5LIB    = /scratch/sdatta/product/10.1.0/agent/perl/lib:/scratch/sdatta/product/10.1.0/agent/perl/lib/site_perl:/scr
EMDROOT     = /scratch/sdatta/product/10.1.0/agent
ORACLE_HOME = /scratch/sdatta/product/10.1.0/Db_1

Fri Aug  5 13:22:55 2005 - Attempting to patch non-Agent Oracle Home...

Fri Aug  5 13:22:55 2005 - OPatch located in the patch...

Fri Aug  5 13:22:55 2005 - ORACLE_HOME has OPatch already present...

Fri Aug  5 13:22:56 2005 - Version of OPatch in ORACLE_HOME is 1.0.0.0.53

Fri Aug  5 13:22:56 2005 - Version of OPatch in the patch is 1.0.0.0.53

Fri Aug  5 13:22:56 2005 - Backing up original OPatch to /scratch/sdatta/product/10.1.0/Db_1/OPatch_1.0.0.0.53...

Fri Aug  5 13:22:56 2005 - Patching OPatch to ORACLE_HOME...

Fri Aug  5 13:22:56 2005 - Patching OPatch to ORACLE_HOME complete. Result = 0

Fri Aug  5 13:22:56 2005 - Patching completed
```

9 CONCLUSION

Oracle Enterprise Manager 10*g* automates software patching greatly thanks to its end to end integration between *MetaLink* and the job subsystem. The centralization of acquiring and distribution not only makes patching more convenient, but also saves cost by a factor of few hundreds. The improvement becomes more pronounced as an enterprise implements a large number of low cost servers and follow the Grid Computing model. In a way, it is more than a product feature. It is rather a service delivered at the console of the Database Administrator, which will fundamentally transform the way patches are delivered, staged and deployed.