

# **Oracle® OPatch**

User's Guide

Release 12.1 for Windows and UNIX

**E39376-01**

June 2013

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	vii
Conventions .....	vii
 <b>1 Introduction to OPatch and Patching</b>	
1.1 Overview of the Patch Process .....	1-2
1.1.1 Obtaining the Patches You Need .....	1-2
1.1.2 Applying the Patch to the Desired Targets .....	1-5
1.1.2.1 Manual Patching .....	1-5
1.1.2.2 Configuration Patching .....	1-6
1.2 Patching with Enterprise Manager .....	1-6
1.3 Who Should Use OPatch? .....	1-6
1.4 What's Covered in this Guide .....	1-7
1.5 OPatch Integration with Other Oracle Software .....	1-7
1.6 How to Access the OPatch Utilities .....	1-8
 <b>2 Binary Patching Using OPatch</b>	
2.1 Obtain the Latest OPatch Utility .....	2-1
2.2 Using OPatch .....	2-1
2.2.1 Patching Workflow .....	2-1
2.2.1.1 Setting the ORACLE_HOME Environment Variable .....	2-2
2.2.1.2 Determining What is Installed On Your System .....	2-2
2.2.1.3 Ensuring Patch Application Prerequisites are Met .....	2-2
2.2.1.4 Applying a Patch .....	2-2
2.2.1.5 Running Post-apply Checks .....	2-3
2.3 Applying a Patch Set Update (PSU) .....	2-3
2.3.1 Applying a Single Patch .....	2-3
2.3.2 Apply Multiple Patches Using a Text File .....	2-3
2.4 Patch Conflict Detection and Resolution .....	2-4
2.4.1 Patch Conflict .....	2-5
2.4.1.1 GI/RAC/DB home Contains More Bug Fixes than the Patch .....	2-5
2.4.1.2 Patch Contains More Bug Fixes than the GI/RAC/DB home .....	2-7
2.4.1.3 GI/RAC/DB Home and the Patch Contain the Same Bug Fixes .....	2-8

2.4.1.4	Patch Contains a Subset of Bug Fixes that are Already in the GI/RAC/DB Home...	2-9
---------	--	-----

### 3 Patch Orchestration Using OPatchauto

3.1	OPatch Automation (OPatchauto) .....	3-1
3.1.1	Supported Target Configurations .....	3-1
3.1.1.1	Shared Versus Non-Shared (GI or RAC) Homes.....	3-2
3.1.1.2	Patch Application Modes .....	3-2
3.2	Running OPatchauto on a Single Node.....	3-3
3.3	OPatchauto Apply .....	3-4
3.3.1	OPatchauto: System Reboot Request .....	3-5
3.4	Sample OPatchauto Patching Steps.....	3-6

### 4 Troubleshooting OPatch

4.1	Error Messages .....	4-1
4.1.1	Obtaining the Latest OPERR.....	4-1
4.1.1.1	Determining Whether OPERR Uses the Latest Message File .....	4-2
4.1.1.2	Obtaining the Latest Message File from MOS.....	4-2
4.1.1.3	Running OPERR with the Latest Message File .....	4-5
4.2	Debugging: Enable Logging and Tracing.....	4-5
4.3	References .....	4-6
4.4	Products and Patch Types Not Supported by OPatch .....	4-6

### 5 Troubleshooting OPatchauto

5.1	OPatchauto Troubleshooting Architecture .....	5-1
5.2	OPatchauto (Use Cases) .....	5-1
5.2.1	OPatch Fails .....	5-1
5.2.2	Rootcrs.pl .....	5-1
5.2.2.1	Rootcrs.pl Prepatch .....	5-2
5.2.2.2	Rootcrs Problem Use Cases.....	5-3
5.2.3	Patchgen .....	5-4
5.2.4	Datapatch .....	5-6
5.3	Common Error Symptoms/Conditions .....	5-7
5.3.1	OPatch Apply ( OPERR Error).....	5-7
5.3.2	Rootcrs.pl Postpatch.....	5-7
5.3.3	Patcherr .....	5-7
5.4	OPatchauto Log Files.....	5-8
5.4.1	Log File Naming Conventions.....	5-8
5.4.2	Log File Locations.....	5-8

### A OPatch Syntax and Commands

A.1	OPatch Syntax .....	A-1
A.2	apply .....	A-2
A.3	compare.....	A-4
A.4	lsinventory.....	A-5
A.5	lspatches.....	A-6

A.6	napply .....	A-7
A.7	nrollback .....	A-10
A.8	rollback .....	A-12
A.9	query .....	A-14
A.10	version .....	A-15
A.11	prereq .....	A-16
A.12	util .....	A-19

## **B OPatchauto Syntax and Commands**

B.1	OPatchauto Commands .....	B-1
B.1.1	apply .....	B-1
B.1.2	resume .....	B-4
B.1.3	rollback .....	B-4
B.1.4	version .....	B-6

## **Glossary**

## **Index**



---

---

# Preface

The *OPatch User's Guide* provides information about Oracle's patching solutions to help ensure your Oracle products stay current and secure.

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

This document is intended for administrators who want to set up and manage Enterprise Manager security.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website  
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For the latest releases of these and other Oracle documentation, check the Oracle Technology Network at:

<http://www.oracle.com/technetwork/documentation/index.html#em>

Oracle Enterprise Manager also provides extensive Online Help. Click **Help** at the top of any Enterprise Manager page to display the online help window.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.



---

# Introduction to OPatch and Patching

OPatch consists of patching utilities that help ensure your Oracle software stays current and secure. The utilities are:

- **OPatch:** A Java-based utility that enables the application and rollback of patches to Oracle software.
- **OPatchauto:** A patch orchestration tool that generates patching instructions specific to your target configuration and then uses OPatch to perform the patching operations without user intervention. Specifically, OPatchauto can:
  1. Perform pre-patch checks.
  2. Apply the patch
  3. Perform post-patch checks.
  4. Roll back patches when patch deinstallation is required.

These utilities provide you with the flexibility to analyze, troubleshoot, and patch an individual GI (Grid Infrastructure)/RAC (Real Application Cluster)/DB Home environments.

For large-scale IT environments, patching individual GI/RAC/DB homes may not be practical since patching large numbers of targets manually is both monotonous and error prone. To maintain and deploy Oracle patches across many targets across your organization, you can use Enterprise Manager Cloud Control's patch automation capability. For more information about Enterprise Manager's patch management solution, see the "Patching Software Deployments" in the *Oracle® Enterprise Manager Lifecycle Management Administrator's Guide*.

This chapter covers the following introductory and overview topics:

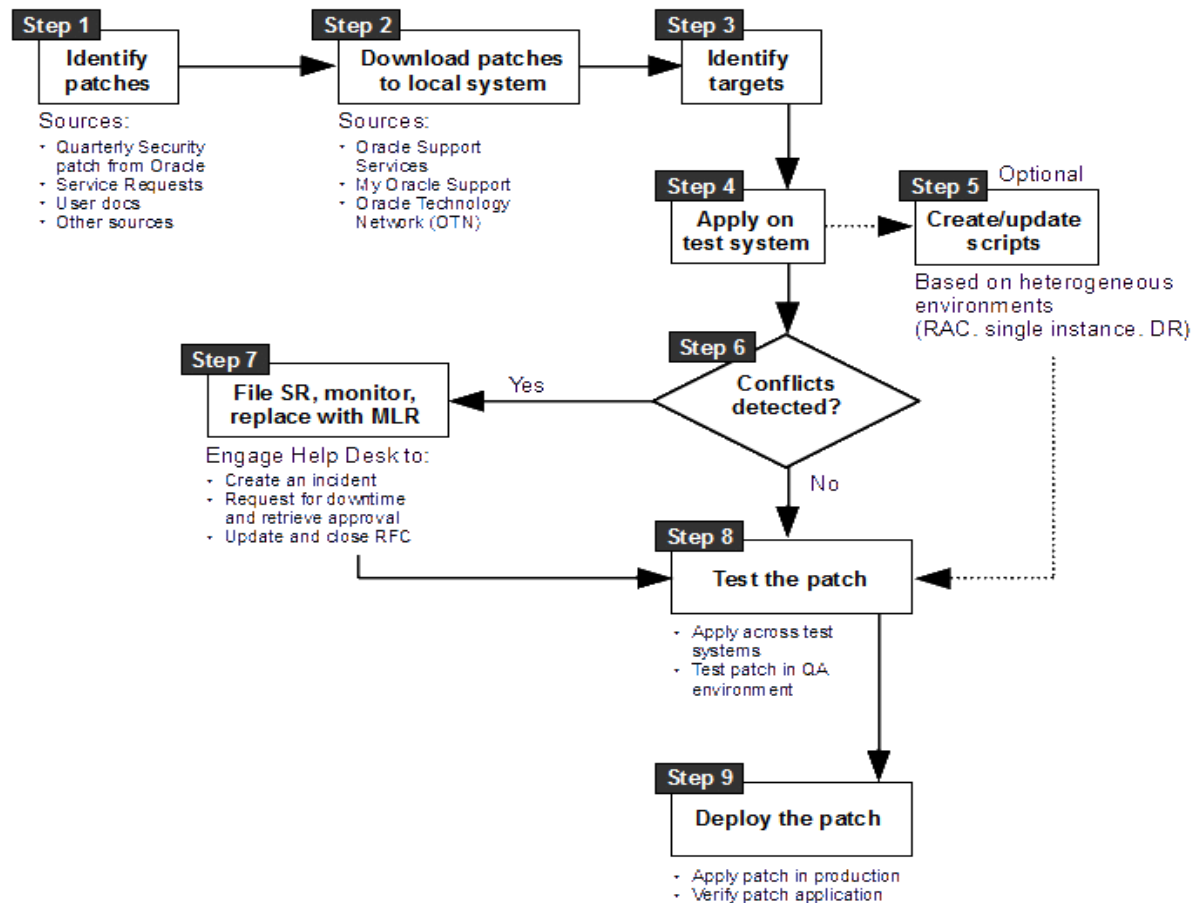
- [Overview of the Patch Process](#)
- [OPatch Integration with Other Oracle Software](#)
- [Who Should Use OPatch?](#)
- [How to Access the OPatch Utilities](#)

The patch process is not always straight forward as there are numerous factors that determine which software patches you need and how these patches should be applied. For example, the types of Oracle software installed on each target, software versions, or platforms on which the software is running are just a few.

## 1.1 Overview of the Patch Process

Regardless of your environment's patching requirements, the basic patching methodology is the same. The normal patching workflow can be broken down into the following nine steps shown in the following figure.

**Figure 1–1 Patch Process Overview - Process Flow**



### 1.1.1 Obtaining the Patches You Need

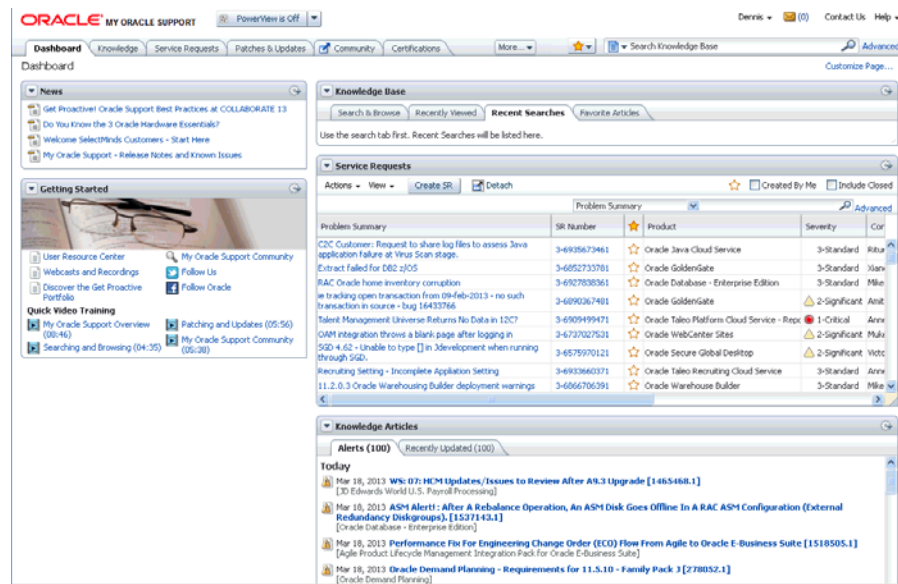
As shown in Figure 1–1, "Patch Process Overview - Process Flow", the first step is to determine what patches you need. You may find out about required patches from blogs, Oracle Technology Network (OTN), Service Requests, Knowledge Articles, Oracle documentation, or any number of other sources. However, the single source of truth for patching is the Oracle Support Web site—My Oracle Support (MOS).

<https://support.oracle.com>

From here, you have access to interactive support tools and information that simplify searching for and obtaining the requisite patches for your Oracle environment. You can find complete documentation about MOS at the following location:

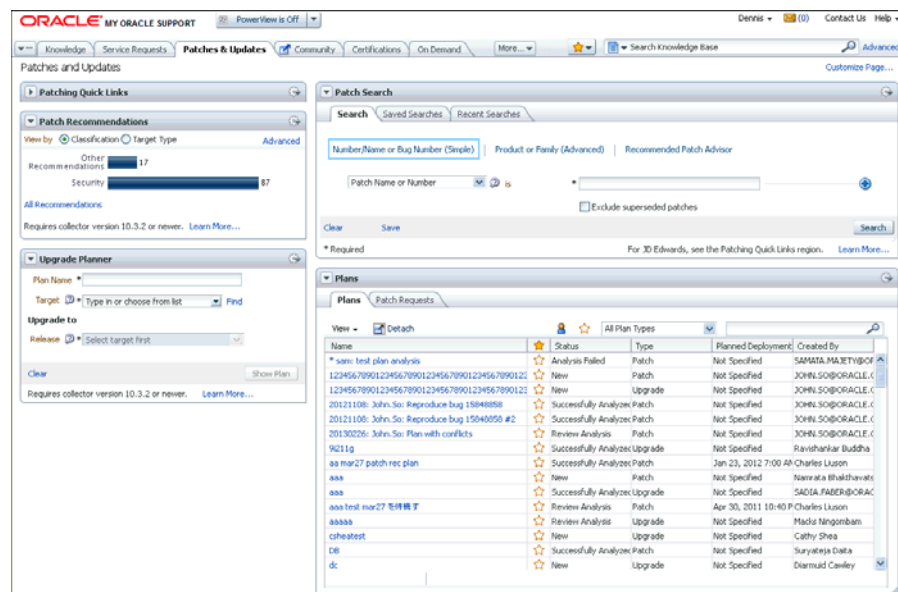
[http://docs.oracle.com/cd/E25290\\_01/index.htm](http://docs.oracle.com/cd/E25290_01/index.htm)

**Figure 1-2 My Oracle Support Main Page**



My Oracle Support contains many features and capabilities that are grouped under tabs across the top of the application. Of primary interest is the **Patches and Updates** tab shown in the following figure.

**Figure 1–3 MOS Patches and Updates**



From this page, you can search for the desired patch based on a specific configuration. One particularly useful search feature is the Recommended Patch Advisor. The Recommended Patch Advisor lets you find recommended and mandatory patches for standalone products, product combinations, or products for a product stack. For example, using the Recommended Patch Advisor, you search for patches for the following product:

- Product: Oracle Database

- Release: 11.2.0.2.0
- Platform: Linux x86-64

This search returns the following results:

Patch Name	Description	Release	Platform (Language)	Classification	Product/Family	Updated	Size	Download Access
14727315	DATABASE PATCH SET UPDATE 11.2.0.2.9 (INCLUDES CPUJAN2013) (Patch)	11.2.0.2.0	Linux x86-64 (American English)	Security	Oracle Database Family	2+ months ago	35.1 MB	Software
14841437	DATABASE SECURITY PATCH UPDATE 11.2.0.2.0 (CPUJAN2013) (Patch)	11.2.0.2.0	Linux x86-64 (American English)	Security	Oracle Database Family	2+ months ago	7.3 MB	Software

By clicking on patch 14727315 (PSU) you are taken to the patch page where you can view bugs resolved by this patch, related Knowledge Articles, or view a generic patch README.

**Patch 14727315: DATABASE PATCH SET UPDATE 11.2.0.2.9 (INCLUDES CPUJAN2013)**

Last Updated: Jan 14, 2013 4:04 AM (2+ months ago) | Size: 35.1 MB

Product: Oracle Database Family | Download Access: Software  
 Release: Oracle 11.2.0.2.0 | Classification: Security  
 Platform: Linux x86-64 | Patch Tag: All Database

**Bugs Resolved by This Patch**

10013177	FUNCTIONAL INDEX CAUSES MAX DECODE GROUP BY SQL TO TRUNCATE VALUES
10013431	LOG4: DATABASE OPEN HANGS AT END OF DBCA CREATION, ORA-32701
10018789	DBMS-BSDB: SPIN IN KGLLOCK CAUSED DR HUNG AND HIGH LIBRARY CACHE LOCK
10019218	ASM DROPPED DDOS BEFORE DISK, JEDPAR_TIME EXPIRED
10021032	NODES REBOOT WHEN MOUNTING ASM DISK CONTAINING VOTE AND OCR
10021111	IN MTS ENV, SESSION USING MIGRATABLE TYP AND REMOTE OPERATION CAN RAISE INT. ERR
10022980	DISK NOT EXPULSED WHEN COMPACT DISABLED
10026601	APPLY ABORTING THIS DUE TO INCORRECT OLD COLUMN VALUE, BUT COLUMN VALUES DO MATCH
10031806	MR: EFFECTIVE RM MAX CPU CARRYING IS WRONG WITH LARGE NUMBER OF CPUS
10057377	A CORRUPT BLOCK WAS DETECTED DURING SHUTDOWN

Open Readline to View All Bugs

**Related Knowledge to this Patch**

14727315.8	Bug 14727315 - 11.2.0.2.9 Database Patch Set Update (PSU)	Modified 01/14/2013
756671.1	Oracle Recommended Patches - Oracle Database	Modified 01/14/2013
1061295.1	Patch Set Updates - One-off Patch Conflict Resolution	Modified 02/28/2013
1454618.1	Quick Reference to Patch Numbers for Database PSU, SPU, CPU, Bundle Patches and Patchsets	Modified 03/06/2013
763736.1	Quick Reference to RDBMS Database Patchset Patch Numbers	Modified 02/28/2013
1314319.1	Bug Fix List: the 11.2.0.2 Patch Bundles for Oracle Database Machine	Modified 02/28/2013
1340011.1	11.2.0.2 Patch Set Updates - List of Psets in each PSU	Modified 02/28/2013
14275621.0	Bug 14275621 - 11.2.0.2.0 Database Patch Set Update (PSU)	Modified 01/14/2013
1179474.1	11.2.0.2 Patch Set - Availability and Known Issues	Modified 01/14/2013
1907461.1	Patch Set Update and Critical Patch Update: December 2013 Availability Document	Modified 01/03/2013

**Download** | **Read Me** | **Add to Plan**

All-time Downloads: No history available | View Trends

**Be the First!**  
 Be the first to share your experience installing this patch.  
[Start a Discussion](#)

Learn More about the Patch Community

From this page, you also complete **Step 2** of the patching workflow—Download the patch to your local system. The following list summarizes sources from which you can obtain patches.

- **Oracle Support Services:** If you are working directly with an Oracle Support engineer, you may be provided with a diagnostic patch or an interim patch.
- **My Oracle Support:** As part of your regular patch maintenance schedule, you can obtain all patches from My Oracle Support:

<https://support.oracle.com>

Once you log in, click the **Patches & Updates** tab to begin your patch search. My Oracle Support offers several patch download options and automated tools to help you keep current with patches. See the Patches & Updates Web-based help for more information:

[http://docs.oracle.com/cd/E25290\\_01/doc.60/e25224/patchesupdates.htm#CJAGJJGI](http://docs.oracle.com/cd/E25290_01/doc.60/e25224/patchesupdates.htm#CJAGJJGI)

- **Oracle Technology Network:** Some Oracle software may be distributed through the Oracle Technology Network:

<http://www.oracle.com/technetwork/indexes/downloads/index.html>

### Types of Oracle Patches

Oracle regularly makes patches available to upgrade features, enhance security, or fix problems with supported software. The major types of patches are:

- **Interim patches** - contains a single bug fix or a collection of bug fixes provided as required
- **Interim patches for security bug fixes** - contain customer-specific security bug fixes
- **Diagnostic patches** - intended to help diagnose or verify a fix or a collection of bug fixes
- **Bundle Patch Updates (BPUs)** - a cumulative collection of fixes for a specific product or component
- **Patch Set Updates (PSUs)** - a cumulative collection of high impact, low risk, and proven fixes for a specific product or component and Security Patch Updates
- **Security Patch Updates (SPU)** - a cumulative collection of security bug fixes. SPUs were formerly known as Critical Patch Updates (CPU).
- **System Patch** - contains several sub-patches in a format that can be used by OPatchauto.
- **Merge Label Request (MLR)** - a *merge* of two or more fixes. MLR creation requires a *label* for the new set of merged code and a Patch Set Exemption

## 1.1.2 Applying the Patch to the Desired Targets

Now that you have the requisite patch, you determine which targets in your environment need to be patched (Step 3 in the patching workflow) and then apply the patch to each target (Step 4). Step 4 is where the OPatch utilities come into play. See [Figure 1-1, "Patch Process Overview - Process Flow"](#) to view the complete patching workflow.

---

**Note:** Ensure that you have the latest version of OPatch. For more information, see ["Obtain the Latest OPatch Utility"](#) on page 2-1.

---

### 1.1.2.1 Manual Patching

Using OPatch, you follow the generic instructions in the patch README. You can view the patch README bundled with the patch or directly from the MOS page for the patch in question.

As shown in the README excerpt, you are required to read the linked support documentation and fill in the details of your specific configuration before you can implement any of the commands or add them to custom install scripts. Although this

method is laborious, it provides you with a great deal of diagnostic capability and control if patch conflicts arise. See [Chapter 2, "Binary Patching Using OPatch."](#)

### 1.1.2.2 Configuration Patching

OPatchauto performs end-to-end configuration patching. Configuration patching is the process of patching a target based on its configuration. By incorporating the site configuration information into the patch process, OPatchauto is able to simplify patching tasks by automating most of the steps.

See [Chapter 3, "Patch Orchestration Using OPatchauto"](#) for more information.

## 1.2 Patching with Enterprise Manager

As useful as the OPatch utilities are, by themselves, they are limited in their ability to apply patches to large numbers of targets because they patch one GI/RAC/DB home at a time. This could be challenging and time consuming in a large, heterogeneous IT environments.

In order to handle large-scale patching, Oracle provides a new patch management solution that integrates OPatch with Enterprise Manager Cloud Control 12c. Enterprise Manager's tight integration with My Oracle Support (MOS) allows you to view patch recommendations, search patches, and roll out patches from a single user interface. In addition, Enterprise Manager's advanced Patch Plan feature provides you with a complete, end-to-end orchestration of the patching workflow. Automating the selection of deployment procedures and analysis of patch conflicts greatly reduces manual effort required to patch complex IT environments.

Enterprise Manager integrates both OPatch and My Oracle Support for downloading and applying patches. See the following documentation for information:

- See "Part VII: Patch Management" in the *Oracle® Enterprise Manager Lifecycle Management Administrator's Guide*:  
[http://docs.oracle.com/cd/E24628\\_01/em.121/e27046/part\\_patching.htm#BGBJFJBG](http://docs.oracle.com/cd/E24628_01/em.121/e27046/part_patching.htm#BGBJFJBG)
- See the "Patching Enterprise Manager" chapter in the *Oracle® Enterprise Manager Cloud Control Administrator's Guide*:  
[http://docs.oracle.com/cd/E24628\\_01/doc.121/e24473/patching.htm#CHDEBABC](http://docs.oracle.com/cd/E24628_01/doc.121/e24473/patching.htm#CHDEBABC)
- Automation of patch conflict resolution and deployment through EM 12c. (Webcast)  
<https://oracleaw.webex.com/oracleaw/lshr.php?AT=pb&SP=EC&rID=74843197&rKey=20438fac9f950a34>

## 1.3 Who Should Use OPatch?

While OPatch is integrated with many Oracle product installations, you may still find that you need to use the OPatch directly. The core OPatch tool is used directly by admins as part of manual patching. OPatchauto invokes OPatch, so an understanding of core OPatch is useful. OPatch is also useful for conflict detection and resolution as well as troubleshooting.

---

---

**Note:** Before performing any patch task, *always* read the patch README file for any special patching instructions.

---

---

You can use the OPatch utilities if your administrative tasks require you to:

- Report on installed products and patches.
- Apply one or more patches.
- Roll back the application of one or more patches.
- Detect conflicts among incoming patches and between it and previous patches that have been applied. OPatch suggests the best options to resolve a conflict.

## 1.4 What's Covered in this Guide

This document describes how to use these patching utilities and covers the following topics:

---

---

**Note:** Recommendation of what tool to use when is stated in the README of the given patch. Always start with the README.

---

---

- [Chapter 2, "Binary Patching Using OPatch"](#) - Describes the basic functions of the core opatch tool, that applies patches to an Oracle Home.
- [Chapter 3, "Patch Orchestration Using OPatchauto"](#) - Recommended for administrators who wish to apply GI-RAC or Exadata patches to GI node in one shot via an orchestration tool.
- [Chapter 4, "Troubleshooting OPatch."](#)
- [Chapter 5, "Troubleshooting OPatchauto"](#)
- [Appendix A, "OPatch Syntax and Commands"](#)
- [Appendix B, "OPatchauto Syntax and Commands"](#)

With these patching tools, you can design and implement a patch plan based on the configuration of your Oracle products.

---

---

**Note:** Before patching any Oracle product, always check the product documentation for patching instructions.

---

---

## 1.5 OPatch Integration with Other Oracle Software

In addition to Enterprise Manager, many Oracle software products have integrated the OPatch utilities to provide for a seamless and efficient patching task. Depending on the application, the call to the OPatch utility may be transparent, and all patching activity is maintained within the respective application.

These applications listed below have integrated OPatch into their respective environments. Always check the user documentation for any patching instructions before applying a patch.

### Fusion Middleware/Fusion Applications

Other Oracle products, such as Fusion Middleware and Fusion Applications, integrate OPatch and may require different interaction to apply a particular patch. Refer to the following documentation:

- For patching Fusion MiddleWare:

- Section 2.3 "OPatch in a Fusion Middleware Environment" in the *Oracle® Fusion Middleware Patching Guide*:  
[http://docs.oracle.com/cd/E23943\\_01/doc.1111/e16793/opatch.htm#PATCH159](http://docs.oracle.com/cd/E23943_01/doc.1111/e16793/opatch.htm#PATCH159)
- Section 3 Applying the Latest Oracle Fusion Middleware Patch Set in the *Oracle® Fusion Middleware Patching Guide*:  
[http://docs.oracle.com/cd/E23943\\_01/doc.1111/e16793/patch\\_set\\_installer.htm](http://docs.oracle.com/cd/E23943_01/doc.1111/e16793/patch_set_installer.htm)
- For patching Fusion Applications:
  - Section 3 Using Oracle Fusion Applications Patch Manager in the *Oracle® Fusion Applications Patching Guide*:  
[http://docs.oracle.com/cd/E29505\\_01/fusionapps.1111/e16602/applypatches.htm#CIHBDCBE](http://docs.oracle.com/cd/E29505_01/fusionapps.1111/e16602/applypatches.htm#CIHBDCBE)

## 1.6 How to Access the OPatch Utilities

With OPatch integrated in many Oracle products, the utility is automatically installed when you install the respective product (for example, Enterprise Manager). The patching tools are installed in the following directories:

- OPatch - `$ORACLE_HOME/OPatch/opatch`
- OPatchauto - `$ORACLE_HOME/OPatch/patchauto`

See [Appendix A, "OPatch Syntax and Commands,"](#) and [Appendix B, "OPatchauto Syntax and Commands"](#) for a complete list of commands and options supported by OPatch utilities.



---

## Binary Patching Using OPatch

OPatch is a utility that allows you to apply and/or roll back interim patches to Oracle's software. The manual process of applying a patch is called *binary patching*. For bits patching, you can use the OPatch utility to:

- [Obtain the Latest OPatch Utility](#)
- [Using OPatch](#)
- [Applying a Patch Set Update \(PSU\)](#)
- [Patch Conflict Detection and Resolution](#)

---

**Note:** Always refer to the patch README for any special instructions before you apply a patch.

Oracle recommends that you back up the ORACLE\_HOME before any patch operation. You can back up the ORACLE\_HOME using your preferred method. You can use any method, such as `zip`, `cp -r`, `tar`, and `cpio` to compress the ORACLE\_HOME.

---

### 2.1 Obtain the Latest OPatch Utility

You should use the version of OPatch that supports the ORACLE\_HOME release. For example, if you are patching a 12.0.1 Oracle Home, then use OPatch version 12.0.1. If the Oracle Home is version 11.2, then use OPatch version 11.2.

Oracle recommends that you use the latest released OPatch for 12.1 releases, which is available for download from My Oracle Support (patch 6880880). Select the ARU link for the 12.1.0.1.0 release.

<https://updates.oracle.com/download/6880880.html>

### 2.2 Using OPatch

Before you use the OPatch command and available options, you must check that OPatch prerequisites have been fulfilled.

#### 2.2.1 Patching Workflow

Using the OPatch utility to patch your GI/RAC/DB home typically consists of the following steps:

1. [Setting the ORACLE\\_HOME Environment Variable](#)
2. [Determining What is Installed On Your System](#)

3. [Ensuring Patch Application Prerequisites are Met](#)
4. [Applying a Patch](#)
5. [Running Post-apply Checks](#)

#### 2.2.1.1 Setting the ORACLE\_HOME Environment Variable

OPatch verifies whether the GI/RAC/DB home is present. You must ensure that the ORACLE\_HOME environment variable is set to the GI/RAC/DB home of the product you are trying to patch. Check your respective vendor documentation for the details on how to set the environment variable.

---

---

**Note:** Oracle Universal Installer binaries and inventories must be present in the home to be patched.

---

---

Other environment variables used include:

- OPATCH\_DEBUG — Boolean setting that specifies the amount of logging OPatch should perform.
- PATH — GI/RAC/DB home path information.

---

---

**Note:** Adding \$ORACLE\_HOME/OPatch to the \$PATH makes it more convenient to execute the OPatch commands from any directory.

---

---

#### 2.2.1.2 Determining What is Installed On Your System

The next step is to determine what is already installed on your system. For this, you use the OPatch `lsinventory` command with either the `patch` or `patch_id` options.

##### **Example 2-1** *lsinventory Command*

```
opatch lsinventory
```

For more information about this command, see "[lsinventory](#)" on page A-5.

#### 2.2.1.3 Ensuring Patch Application Prerequisites are Met

After you have determined that your system configuration is appropriate for the patches you wish to apply, it is advisable to view the operations OPatch will perform before performing the patch application to help determine whether all system prerequisites are met before applying the patch. For this, you use the OPatch `-report` option to print all patch application actions OPatch will perform without actually executing the actions.

##### **Example 2-2** *report Option*

```
opatch apply -report
```

For more information about the `-report` option, see "[OPatch Syntax and Commands](#)" on page A-1.

#### 2.2.1.4 Applying a Patch

Once you have determined that the patch can be applied to your system successfully, you can now use OPatch to apply the patch. For this, you use the OPatch `apply` command.

**Example 2-3 `apply` Command**

```
opatch apply /tmp/patch/12345678
```

For more information about this command, see ["apply"](#) on page A-2.

**2.2.1.5 Running Post-apply Checks**

After you have applied the patch to your system, you should perform a final check to ensure all patches have been successfully applied. For this, you again use the OPatch `lsinventory` command with either the `patch` or `patch_id` options. For more information about this OPatch command, see ["lsinventory"](#) on page A-5.

**2.3 Applying a Patch Set Update (PSU)**

Once you have verified the prerequisite checks, use OPatch to apply a patch:

- [Applying a Single Patch](#)
- [Apply Multiple Patches Using a Text File](#)

The OPatch utility is located in the `$Oracle_Home/OPatch` directory. You can run it with various commands and options. See [Appendix A, "OPatch Syntax and Commands,"](#) for a complete list of command options available with OPatch.

**2.3.1 Applying a Single Patch**

You apply a single patch following the generic patching workflow discussed earlier:

1. [Obtain the Latest OPatch Utility](#)
2. [Setting the ORACLE\\_HOME Environment Variable](#)
3. [Determining What is Installed On Your System](#)
4. [Ensuring Patch Application Prerequisites are Met](#)
5. [Applying a Patch](#)
6. [Running Post-apply Checks](#)

Once you have downloaded the patch, you can apply it (step 5) using the following command:

```
# opatch apply <patch directory location>/<patch ID>
```

For example:

```
# opatch apply /tmp/patch/12345678
```

**2.3.2 Apply Multiple Patches Using a Text File**

You can create a text file containing the location of all patches you need to apply. Use OPatch to reference the file and apply the patches:

1. Create the text file of the patch location. The entry should have each patch location on a separate line:

```
vi patches.txt
/tmp/patchlocation1/12345678
/tmp/patchlocation2/12365478
/scratch/patchlocation3/32165487
```

Save your changes.

## 2. Apply the patches with OPatch:

```
# opatch napply -idFile <location of text file>
```

For example:

```
#opatch napply -idFile /tmp/patches/patches.txt
```

## 2.4 Patch Conflict Detection and Resolution

OPatch detects and reports any conflicts encountered when applying a patch with a previously applied patch. The patch application fails in case of conflicts. You can use the `-force` option of OPatch to override this failure. If you specify `-force`, the installer firsts rolls back any conflicting patches and then proceeds with the installation of the desired patch.

You may experience a bug conflict and might want to remove the conflicting patch. This process is known as patch rollback. During patch installation, OPatch saves copies of all the files that were replaced by the new patch before the new versions of these files are loaded, and stores them in `$ORACLE_HOME/.patch_storage`. These saved files are called *rollback files* and are key to making patch rollback possible. When you roll back a patch, these rollback files are restored to the system. If you have a complete understanding of the patch rollback process, you should only override the default behavior by using the `-force` flag. To roll back a patch, execute the following command:

```
$ OPatch/opatch rollback -id <Patch_ID>
```

A patch conflict occurs when multiple fixes in different patches touch the same files but have not been tested together as a single entity. OPatch and OPatchauto help you avoid these conflicts by identifying these conditions. When patch conflicts occur and you are unable to resolve them using documented support procedures, MOS then becomes the go-to source for technical assistance. Conflict resolution may entail filing a Service Request and obtaining a Merge Label Request (MLR) patch to overcome a patching issue. Once a solution has been found, you use OPatch to apply the fixed patch.

### Typical Patching Scenarios

When working with patches, there are four scenarios you may encounter based on the content of the patch versus the condition of the GI/RAC/DB home.

The following table describes these four situations and suggested actions.

GI/RAC/DB Home Versus Patch Content	Description	Action
GI/RAC/DB home > Patch	You have more bug fixes in the GI/RAC/DB home than the patch. You do not need to apply the patch.	Do Nothing
GI/RAC/DB home < Patch	You have fewer bug fixes in the GI/RAC/DB home than the patch. You should apply the patch.	Apply the Patch
GI/RAC/DB home = Patch	The GI/RAC/DB home and the patch have the same bug fixes.	Do Nothing

<b>GI/RAC/DB Home Versus Patch Content</b>	<b>Description</b>	<b>Action</b>
GI/RAC/DB home and Patch Intersect	Under certain circumstances, the decision to apply the patch may not be clear, as would be the case where there are overlapping bug fixes: For example, you have bug fixes 1, 2, 3 and 4 in the GI/RAC/DB home. The patch you want to apply contains bug fixes 3, 4, 5, and 6. In this situation, bug fixes 3 and 4 are in both the GI/RAC/DB home and the patch. If you install the patch, you will lose bug fixes 3 and 4. If you install the patch, you will lose some of the bug fixes.	Call Oracle Support or use the MOS patch plan feature to resolve the conflict.

As shown above, all patches may not be compatible with one another. This is called a conflict situation. OPatch detects such situations and raises an error when it detects a conflict. The following section illustrates in detail the patch/conflict states shown in the table.

## 2.4.1 Patch Conflict

OPatch can detect the following patch conflict conditions:

- Patch Contains More Bug Fixes than the GI/RAC/DB home
- GI/RAC/DB home Contains More Bug Fixes than the Patch
- GI/RAC/DB Home and the Patch Contain the Same Bug Fixes
- Patch Contains a Subset of Bug Fixes that are Already in the GI/RAC/DB Home

### 2.4.1.1 GI/RAC/DB home Contains More Bug Fixes than the Patch

In this case, the GI/RAC/DB home already contains bug fixes in the patch in addition to other bug fixes.

**Recommended Action:** Do nothing.

#### Example

Consider the following scenario:

- Patch A, installed in the GI/RAC/DB home, fixed bugs 1, 2, and 3.
- Patch B, installed in the GI/RAC/DB home, fixed bugs 10, 11, and 12.
- Patch D, to be installed, fixes bugs 1 and 2.

Patch D is a subset of Patch A.

#### Using the skip\_subset Option

When you want to skip patches formerly applied in the GI/RAC/DB home that are now subsets of other patches you want to apply now, you can use the `skip_subset` option of `napply`. For example, if you used `napply` yesterday for patch A that fixed bugs 1 and 2, then you use `napply` today with the `skip_subset` option for patch B that fixes bug 1 and patch C that fixes bugs 1, 2, and 3. Then subset patch A is skipped, and patch C then becomes a superset of patch A.

The following example output shows the message you would see when you use the `skip_subset` flag:

```
$ OPatch/patch napply /<path to patch>/ -id 100,101 -skip_duplicate -skip_subset
```

```
Oracle Interim Patch Installer version 11.1.0.9.10
Copyright (c) 2012, Oracle Corporation. All rights reserved.

Oracle Home      : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_1
Central Inventory : /<path to oraInventory directory>/oraInventory
  from           : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1//oraInst.loc
OPatch version   : 11.1.0.9.10
OUI version      : 11.1.0.7.0
Log file location : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1/cfgtoollogs/opatch/opatch2013-05-14_11-31-29AM_1.log

Verifying environment and performing prerequisite checks...

Conflicts/Supersets for each patch are:

Patch : 100

      Bug SubSet of 101
      Subset bugs are:
      111

Patch : 101

      Bug Superset of 100
      Super set bugs are:
      111

Skip these patches because they are subset of other patches in the list:  100
Proceed with these patches:  101
Checking skip_duplicate
Checking skip_subset

OPatch found that the following patches are not required.
They are either subset of the patches in Oracle Home (or) subset of the patches in
the given list
(or) duplicate :
    100
OPatch continues with these patches:  101

Do you want to proceed? [y|n]
y
User Responded with: Y
All checks passed.
Provide your email address to be informed of security issues, install and
initiate Oracle Configuration Manager. Easier for you if you use your My
Oracle Support Email address/User Name.
Visit http://www.oracle.com/support/policies.html for details.
Email address/User Name:

You have not provided an email address for notification of security issues.
Do you wish to remain uninformed of security issues ([Y]es, [N]o) [N]: Y

Backing up files...
Applying interim patch '101' to OH '/<path to OH
directory>/oraclehome/11.1.0.6/dbhome_1'

Patching component oracle.rdbms.rsfc, 11.1.0.7.0...

Verifying the update...
```

Patches 101 successfully applied.  
 OPatch Session completed with warnings.  
 Log file location: /<path to OH directory>/oraclehome/11.1.0.6/dbhome\_  
 1/cfgtoollogs/patch/patch2013-05-14\_11-31-29AM\_1.log

OPatch completed with warnings.

The expected behavior for the Apply and Napply commands is listed in [Table 2-1](#).

**Table 2-1 Expected Conflict Behavior**

Command	GI/RAC/DB home>Patch	GI/RAC/DB home<Patch	GI/RAC/DB home=Patch	GI/RAC/DB home Intersects Patch
<b>Apply</b>	OPatch performs an automatic rollback, then an apply.	After the merge request, OPatch performs an automatic rollback, then performs an apply.	OPatch performs an automatic rollback, then performs a reapply.	OPatch reports the conflict. After the merge request, OPatch performs an automatic rollback, then an apply.
<b>Napply</b>	OPatch performs an automatic rollback, then an apply.	OPatch reports the subset and skips the subset patch. It then continues and applies the other patches.	OPatch performs an automatic rollback, then a reapply.	OPatch reports the conflict, then asks you to run again without applying a bug conflict patch.  You can use the -force option to instruct OPatch to automatically roll back the conflicting patch, then apply the new patch.

In the case of napply, because multiple patches are being applied, patches may conflict with each other in the same way a patch and an GI/RAC/DB home conflict (OH>Patch, OH<Patch, OH=Patch, and OH intersects Patch). When patch/patch conflicts arise, the same recommended actions should be applied.

The following example applies all patches under the <patch\_location> directory. OPatch skips duplicate patches and subset patches (patches under <patch\_location> that are subsets of patches installed in the GI/RAC/DB home).

#### **Example 2-4**

```
opatch napply <patch_location> -skip_subset -skip_duplicate
```

The following example applies patches 1, 2, and 3 that are under the <patch\_location> directory. OPatch skips duplicate patches and subset patches (patches under <patch\_location> that are subsets of patches installed in the GI/RAC/DB home).

#### **Example 2-5**

```
opatch napply <patch_location> -id 1,2,3 -skip_subset -skip_duplicate
```

### **2.4.1.2 Patch Contains More Bug Fixes than the GI/RAC/DB home**

If a patch contains bug fixes that contain have already been fixed in the GI/RAC/DB home plus additional bug fixes, then this patch is considered a superset of patches already applied to the GI/RAC/DB home. If a patch superset condition is detected, it

is not considered an error situation: All the subset patches are removed from the system and the new patch is applied.

**Recommended Action:** Apply the patch.

### Example

Consider the following scenario:

- Patch A, installed in the GI/RAC/DB home, fixed bugs 1, 2, and 3.
- Patch B, installed in the GI/RAC/DB home, fixed bugs 10, 11, and 12.
- Patch C, to be installed, fixes bugs 1, 2, 3, and 4.

Patch C is considered a superset of Patch A.

### Using the `-no_bug_superset` Flag

If you want OPatch to error out if the current patch bugs-to-fix is a superset or the same as an installed patch bugs-fixed in the GI/RAC/DB home directory, you can use the `-no_bug_superset` flag:

```
$ OPatch/patch apply -no_bug_superset <Path_To_Patch>
```

The following example output shows the message you would see when you use the `-no_bug_superset` flag:

```
$ OPatch/patch apply -no_bug_superset /<path_to_patch>/patch101_CP1/
Oracle Interim Patch Installer version 11.1.0.9.10
Copyright (c) 2012, Oracle Corporation. All rights reserved.

Oracle Home      : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_1
Central Inventory : /<path to oraInventory directory>/oraInventory
   from           : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1//oraInst.loc
OPatch version    : 11.1.0.9.10
OUI version       : 11.1.0.7.0
Log file location : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1/cfgtoollogs/patch/patch2013-05-14_11-34-19AM_1.log

Applying interim patch '101' to OH '/<path to OH
directory>/oraclehome/11.1.0.6/dbhome_1'
Verifying environment and performing prerequisite checks...
Interim patch 101 is a superset of the patch(es) [ 100 ] in the Oracle Home
OPatch system modification phase did not start: Interim patch 101 is a superset of
the patch(es) [ 100 ] in OH /<path to OH directory>/oraclehome/11.1.0.6/dbhome_1
Log file location: /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1/cfgtoollogs/patch/patch2013-05-14_11-34-19AM_1.log

OPatch stopped on request.
```

### 2.4.1.3 GI/RAC/DB Home and the Patch Contain the Same Bug Fixes

In this case, a patch fixes the same set of bugs fixed by another patch. For example, if you applied Patch A that fixed bugs 1, 2 and 3, and now apply Patch B that also fixes bugs 1, 2 and 3, then Patch B is a duplicate of Patch A. A patch is always a duplicate of itself.

### Using the `skip_duplicate` Option

If you specify this option, OPatch removes duplicate patches from the list of patches to be applied. For example, if you used `napply yesterday` for Patch A discussed above,



then use `opatch apply` today with the `-skip_duplicate` option for Patch A and other patches, duplicate Patch A is skipped.

#### 2.4.1.4 Patch Contains a Subset of Bug Fixes that are Already in the GI/RAC/DB Home

In this case, a patch conflict condition exists whereby a set of bugs to be fixed by the current patch intersects with some bugs already fixed by one or more previously installed patches.

**Recommended Action:** Call Oracle Support. You may need to request an MLR patch.

##### Example

The following example illustrates this patch conflict condition during patch application. In the example, `opatch apply` fails with error code 56.

```
$ OPatch/opatch apply /<path to patch directory>/patch100_CP0/
Oracle Interim Patch Installer version 11.1.0.9.10
Copyright (c) 2012, Oracle Corporation. All rights reserved.

Oracle Home           : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_1
Central Inventory     : /<path to oraInventory>/oraInventory
  from                : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1//oraInst.loc
OPatch version        : 11.1.0.9.10
OUI version           : 11.1.0.7.0
Log file location     : /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1/cfgtoollogs/opatch/100_May_14_2013_12_18_10/apply2013-05-14_12-18-09PM_1.log

Applying interim patch '100' to OH '/<path to OH
directory>/oraclehome/11.1.0.6/dbhome_1'
Verifying environment and performing prerequisite checks...
Interim patch 100 is a subset of the patch(es) [ 101 ] in OH /<path to OH
directory>/oraclehome/11.1.0.6/dbhome_1.
All the fixes of this patch 100 is already present in the Oracle Home. No need to
apply this patch.
Log file location: /<path to OH directory>/oraclehome/11.1.0.6/dbhome_
1/cfgtoollogs/opatch/100_May_14_2013_12_18_10/apply2013-05-14_12-18-09PM_1.log

Recommended actions: The fixes by this patch are currently in the Oracle Home.
There is no need to apply this patch.
```

**OPatch failed with error code 56**



---

## Patch Orchestration Using OPatchauto

Patch orchestration is the automated execution of the patching steps, such as the execution of pre-patch checks, stopping services, applying the binary patches, and starting the services. Patch orchestration for Oracle Database 12c applies the patch to the GI/RAC configuration on that machine, including all of its databases. The OPatchauto patch orchestration utility has been made available with version 12.1 of the OPatch utility.

This chapter covers the following topics:

- [OPatch Automation \(OPatchauto\)](#)
- [Running OPatchauto on a Single Node](#)
- [OPatchauto Apply](#)
- [Sample OPatchauto Patching Steps](#)

---

**Important:** This chapter applies to Oracle Database 12c only.

---

### 3.1 OPatch Automation (OPatchauto)

With OPatchauto, you can automatically patch the typical Grid Infrastructure (GI) and RAC home directories with minimal intervention. [Figure 3–1](#) shows an overview flow of the patch process using OPatchauto.

OPatchauto performs many of the pre-patch checks (see ["Using OPatch"](#) on page 2-1) as well as the post-patch verification. The power of OPatchauto lies in its ability to perform end-to-end *configuration patching*. Configuration patching is the process of patching a GI or RAC home based on its configuration. By incorporating the configuration information into the patch process, OPatchauto streamlines patching tasks by automating most of the steps.

OPatchauto uses your GI/RAC configuration and, from that information, automatically generates patching instructions specific to your site configuration. OPatchauto then uses OPatch to implement these instructions and perform the actual application of the patch.

#### 3.1.1 Supported Target Configurations

OPatchauto can be applied to the following general configurations:

- GI Home Shared
- GI Home Not Shared
- RAC Home Shared

- RAC Home Not Shared
- Single-Instance High Availability (SIHA) Home (single node)

### 3.1.1.1 Shared Versus Non-Shared (GI or RAC) Homes

In a shared GI/RAC/DB home, all nodes in the cluster use the same physical copy of the software. This simplifies configuration and management of many database operations because there is a single Home location rather than separate Homes on each node.

When a GI Home or RAC Home is shared, individual nodes within the GI or RAC environments share a single file system and utilize a cluster file system such as Oracle Cluster File System 2 (OCFS2), in addition to sharing the same Home. Although this configuration is more disk space-efficient, the process of patching becomes a bit more complicated as the different nodes are utilizing the same resources/disk space.

---

**Note:** GI shared home installations can be patched only in nonrolling mode

---

In a non-shared GI/RAC/DB home, sometimes referred to as Private GI/RAC/DB home, each node in the cluster maintains a complete copy of the Oracle software tree on local storage. This is the most common way in which Oracle Grid Infrastructure and Real Application Clusters are installed.

The configuration differences between shared and non-shared Homes come into play when determining the patching mode in which OPatchauto is used. See [Section 3.1.1.2, "Patch Application Modes."](#)

### 3.1.1.2 Patch Application Modes

OPatchauto supports two modes of patching a GI or RAC Home - Rolling and Non-rolling. When a patching session is started off (on the first node), the stack has to be up and running on this node. This applies to both rolling and non-rolling modes of patching.

**Rolling Mode (Default Mode):** When performing patching in Rolling mode, a node is shut down, the patch is applied, then the node is brought back up again. This process is repeated for each node in the GI or RAC environment until all nodes are patched. This is the most efficient mode of applying an interim patch to an Oracle RAC setup because this results in no downtime. Not all patches can be applied using Rolling mode. Whether or not a patch can be applied in this way generally specified in the patch metadata. The node (GI Home) from which the `opatchauto` command is executed is considered the LOCAL node and all other nodes are considered REMOTE nodes.

When you begin a rolling mode session, at least 1 remote node has to be up and running.

OPatchauto applies patches in rolling mode by default.

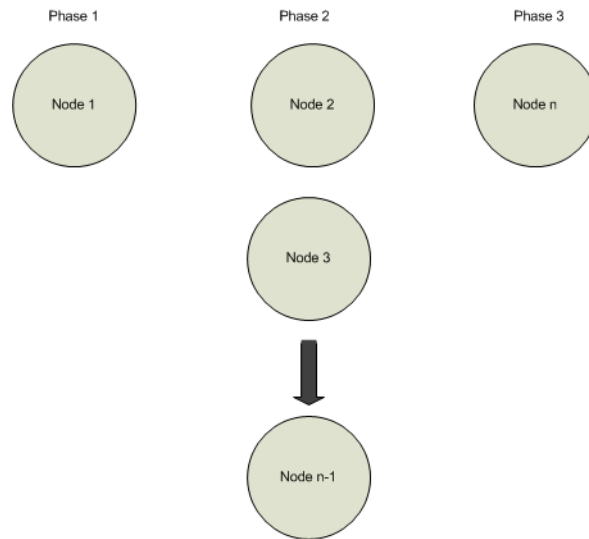
**Non-rolling Mode:** Prior to 12c, a non-rolling upgrade was defined as shutting down Oracle processes on all nodes. Beginning with 12c, non-rolling patching requires the GI stack to be up on local node. The patching operation on first and last node have special steps to perform hence the operation needs to be handled separately but not in parallel with other nodes. The non-rolling patching can be described as three phases:

Beginning with 12c, non-rolling patching occurs in three phases:

1. Patch Node 1
2. Patch Node 2 through  $n-1$
3. Patch Node  $n$

When you start a non-rolling mode session none of the remote nodes can up and running: All nodes must be stopped.

As shown in the following figure, given  $n$  nodes, you begin the non-rolling patch session by patching a single node, then patch nodes two through  $n-1$  in parallel, and finally patch node  $n$  to finish the patching session.



To run OPatchauto in non-rolling mode, you must explicitly specify the `-nonrolling` option.

#### Node Availability at the Start of the Patching Session

In order to start a new patching session, the following conditions must be met.

- Local node must be **up** for both Rolling and Non-rolling modes.
- At least one of the remote nodes must be **up** in order to start a Rolling mode session.
- ALL the remote nodes must be **down** in order to start a Non-rolling session.

#### Patch Application Mode Conflict

As mentioned earlier, OPatchauto applies patches in rolling mode by default. If the patch is applied in rolling mode but the patch content is not rollable (content does not support application in rolling mode), OPatchauto will error out when attempting to run `rootcrs.pl -prepatch`.

## 3.2 Running OPatchauto on a Single Node

To ensure successful patching, Oracle recommends that a patching session always consist of the four following steps:

1. Run `opatch lsinventory`

**When to run:** During cluster uptime.

Before you begin a new patching session you must ensure that all nodes of the cluster are at the same patch level.

```
$GRID_HOME/OPatch/opatch lsinventory
```

#### Example Output

```
=====
Patch level status of Cluster nodes :
Patching Level      Nodes
-----
12345678            node123, node456
```

2. Run `opatchauto apply -analyze`

**When to run:** During cluster uptime.

The `apply -analyze` command simulates an `opatchauto apply` session by running all prerequisite checks, when possible, without making changes to the system (either bits or configurations).

3. Run `opatchauto apply`

**When to run:** During cluster downtime.

The `apply` command applies all patches within a specified system patch to the GI or RAC home from which `opatchauto` is command is run.

4. Run `opatch lsinventory`

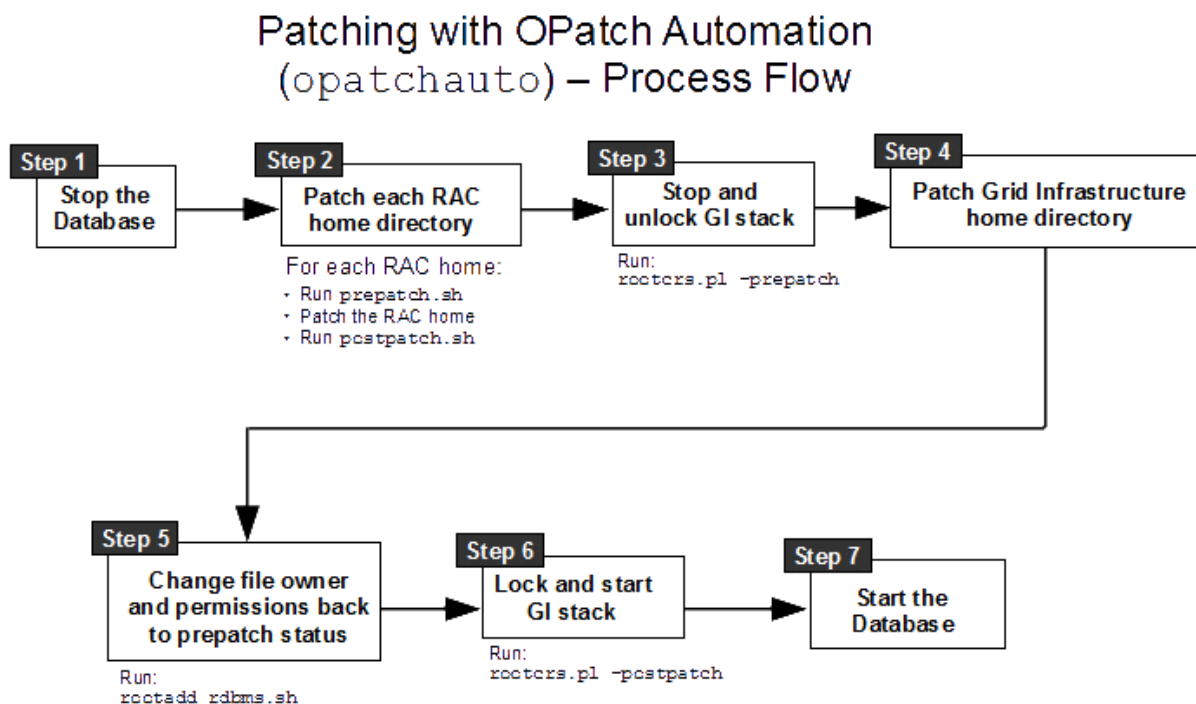
**When to run:** During cluster uptime.

After successfully running `opatchauto apply` on each individual node, you must run `opatch lsinventory` on that node to confirm that the patch level has been correctly updated.

## 3.3 OPatchauto Apply

When you run OPatchauto's `apply` command, numerous operations are performed to implement the complete patch application cycle. These operations vary depending on the environment to be patched. The following environment is representative of the vast majority of patching environments in which OPatchauto is used. For example, a typical patching environment would be one GI Home managing two RAC Homes. When you run `opatchauto apply`, OPatchauto will perform the operations shown in [Figure 3-1, "OPatchauto Apply Process Flow"](#).

Figure 3–1 OPatchauto Apply Process Flow



### 3.3.1 OPatchauto: System Reboot Request

Depending on the patch or the home directory configuration, you may encounter a request to reboot the system. After a reboot during the patching process, you need to invoke the `opatchauto` utility again so that it seamlessly continues with rest of the patch application process..

Typically an error message, as shown in the following example, will be displayed when a problem arises.

#### Example 3–1 OPatchauto Console Error

```
# OPatch/opatchauto apply /scratch/aim1/sh/RDBMS_12.1.0.1.0_LINUX.X64_
130418/patches/v2/nosql/gipsu/11111111 -ocmrf /tmp/ocm.rsp
OPatch Automation Tool
Copyright (c) 2013, Oracle Corporation. All rights reserved.
...
[WARNING] The local database instance 'norcl2' from
'/scratch/aim1/DB12N/app/aim1/product/12.1.0/dbhome_1' is not running, the
following command will NOT be executed:
/bin/bash -c 'ORACLE_HOME=/scratch/aim1/DB12N/app/aim1/product/12.1.0/dbhome_1
ORACLE_SID=$(/scratch/aim1/DB12N/app/aim1/product/12.1.0/dbhome_1/bin/srvctl
status instance -d norcl -n slc00epi | cut -d " " -f 2)
/scratch/aim1/DB12N/app/aim1/product/12.1.0/dbhome_1/OPatch/datapatch'.
To apply the SQL changes patch all the nodes, bring up the database instance and
run this command manually from any one node (run as aim1).
```

**CLSRSC-400: A system reboot is required to continue installing.**

...

Apply Summary:

Following patch(es) are successfully installed:

```
GI_HOME=/u01/GI12/app/12.1.0/grid:13852018, 22222222, 123456788
DB_HOME=/scratch/aime/DB12_2/app/aime/product/12.1.0/dbhome_1:13852018, 123456788
DB_HOME=/scratch/aime1/DB12N/app/aime1/product/12.1.0/dbhome_1:13852018, 123456788
```

**opatchauto failed with error code 1.**

When you receive an error like this, follow the reboot instructions specified in the console. [Example 3–2, "Rebooting the System"](#) shows a system reboot request issued by the user.

### **Example 3–2 Rebooting the System**

```
# OPatch/opatchauto resume -reboot
OPatch Automation Tool
Copyright (c) 2013, Oracle Corporation. All rights reserved.

OPatchauto version : 12.1.0.1.1
OUI version        : 12.1.0.1.0
Running from       : /u01/GI12/app/12.1.0/grid
Log file location  :
/u01/GI12/app/12.1.0/grid/cfgtoollogs/patch/opatch/opatch2013-05-16_13-36-59PM_1.log

OPatchauto will attempt to resume from reboot patching session. This might take
several minutes...

Command "/usr/bin/perl /u01/GI12/app/12.1.0/grid/crs/install/rootcrs.pl
-postpatch" is successfully resumed.
Command "/scratch/aime1/DB12N/app/aime1/product/12.1.0/dbhome_1/bin/srvctl start
home -o /scratch/aime1/DB12N/app/aime1/product/12.1.0/dbhome_1 -n slc00epi -s
/scratch/aime1/DB12N/app/aime1/product/12.1.0/dbhome_
1/OracleHome-50b8f1a0-e220-4b8e-98d7-49177979991f.stat " is successfully resumed.
Command "/scratch/aime/DB12_2/app/aime/product/12.1.0/dbhome_1/bin/srvctl start
home -o /scratch/aime/DB12_2/app/aime/product/12.1.0/dbhome_1 -n slc00epi -s
/scratch/aime/DB12_2/app/aime/product/12.1.0/dbhome_
1/OracleHome-58232a10-3130-4930-b588-0c8594cf8c87.stat " is successfully resumed.
OPatchauto was able to resume from the previous reboot patching session and
complete successfully.

opatchauto succeeded.
```

## **3.4 Sample OPatchauto Patching Steps**

The following example shows the commands OPatchauto runs during a normal patching sessions. The environment being patched consists of a single GI Home (grid) and a single RAC Home (dbhome\_1).

### **Example 3–3 OPatchauto Commands**

```
/u01/GI/app/12.1.0/grid/OPatch/opatch version -oh /u01/GI/app/12.1.0/grid
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -v2c 12.1.0.1.0

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/opatch version -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -v2c 12.1.0.1.0

/u01/GI/app/12.1.0/grid/OPatch/opatch prereq CheckComponents -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/123456788
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid
```



```

/u01/GI/app/12.1.0/grid/OPatch/patch prereq CheckComponents -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/13852018
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid

/u01/GI/app/12.1.0/grid/OPatch/patch prereq CheckComponents -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/22222222
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch prereq
CheckComponents -ph /scratch/aime/sh/RDBMS_MAIN_LINUX.X64_
120801/patches/sql/13852018/123456788 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch prereq
CheckComponents -ph /scratch/aime/sh/RDBMS_MAIN_LINUX.X64_
120801/patches/sql/13852018/13852018_rac -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

/u01/GI/app/12.1.0/grid/OPatch/patch prereq CheckConflictAgainstOH -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/123456788
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid

/u01/GI/app/12.1.0/grid/OPatch/patch prereq CheckConflictAgainstOH -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/13852018
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid

/u01/GI/app/12.1.0/grid/OPatch/patch prereq CheckConflictAgainstOH -ph
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/22222222
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch prereq
CheckConflictAgainstOH -ph /scratch/aime/sh/RDBMS_MAIN_LINUX.X64_
120801/patches/sql/13852018/123456788 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch prereq
CheckConflictAgainstOH -ph /scratch/aime/sh/RDBMS_MAIN_LINUX.X64_
120801/patches/sql/13852018/13852018_rac -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

rm -f /tmp/OracleHome-add73cd4-55c2-4415-807c-cc510d5c508c.stat

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/bin/srvctl stop home -o
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1 -n slc00eph -s
/tmp/OracleHome-add73cd4-55c2-4415-807c-cc510d5c508c.stat

/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/13852018_
rac/custom/scripts/prepatch.sh -dbhome
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch napply -local
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/123456788
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1 -silent -ocmrf /tmp/ocm.rsp

```

```
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/patch napply -local
/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/13852018_rac
-invPtrLoc /u01/GI/app/12.1.0/grid/oraInst.loc -oh
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1 -silent -ocmrf /tmp/ocm.rsp

/scratch/aime/sh/RDBMS_MAIN_LINUX.X64_120801/patches/sql/13852018/13852018_
rac/custom/scripts/postpatch.sh -dbhome
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1

/usr/bin/perl /u01/GI/app/12.1.0/grid/crs/install/rootcrs.pl -prepatch

/u01/GI/app/12.1.0/grid/OPatch/patch napply -local /scratch/aime/sh/RDBMS_MAIN_
LINUX.X64_120801/patches/sql/13852018/123456788 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid -silent -ocmrf
/tmp/ocm.rsp

/u01/GI/app/12.1.0/grid/OPatch/patch napply -local /scratch/aime/sh/RDBMS_MAIN_
LINUX.X64_120801/patches/sql/13852018/13852018 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid -silent -ocmrf
/tmp/ocm.rsp

/u01/GI/app/12.1.0/grid/OPatch/patch napply -local /scratch/aime/sh/RDBMS_MAIN_
LINUX.X64_120801/patches/sql/13852018/22222222 -invPtrLoc
/u01/GI/app/12.1.0/grid/oraInst.loc -oh /u01/GI/app/12.1.0/grid -silent -ocmrf
/tmp/ocm.rsp

/u01/GI/app/12.1.0/grid/rdbms/install/rootadd_rdbms.sh

/usr/bin/perl /u01/GI/app/12.1.0/grid/crs/install/rootcrs.pl -postpatch

/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/bin/srvctl start home -o
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1 -n slc00eph -s
/tmp/OracleHome-add73cd4-55c2-4415-807c-cc510d5c508c.stat

ORACLE_SID=$(/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/bin/srvctl status
instance -d orcl -n slc00eph | cut -d ' ' -f 2)
/scratch/aime/DB12/app/aime/product/12.1.0/dbhome_1/OPatch/datapatch
```

---

## Troubleshooting OPatch

This chapter describes common problems with patching and troubleshooting with OPatch. The following sections are discussed:

- [Error Messages](#)
- [Debugging: Enable Logging and Tracing](#)
- [References](#)
- [Products and Patch Types Not Supported by OPatch](#)

### 4.1 Error Messages

If there are problems with applying a patch, OPatch will return an error message. OPatch provides an integrated error analysis tool, `operr`, that lists both the cause and recommended action to resolve the issue. The tool can be found at the following location:

```
$ORACLE_HOME/OPatch/operr
```

Syntax:

```
operr <error code>
```

The error information file used by the `operr` tool is the latest at the time of release. However, the information is constantly being updated. You can obtain the latest information file from MOS. See [Section 4.1.1, "Obtaining the Latest OPERR"](#) for more information.

#### If OPatch Fails to Run

If OPatch itself does not run, there are two possibilities:

- Oracle Home is not set  
Recommended Action: Set your Oracle Home environment variable.
- JRE/JDK cannot be found  
Recommended Action: Run `opatch apply` using the `-jre` option, which lets you specify the location of an installed JRE/JDK.

#### 4.1.1 Obtaining the Latest OPERR

In this example, you are attempting to apply a patch and OPatch encounters an error condition. You find that the message file is not the latest and must obtain the latest message patch file using My Oracle Support (MOS).

In the following example, OPatch displays an error message with error code 115.

**Example 4–1 OPatch Error Message**

```
bash-4.1$ OPatch/patch apply /scratch/my_admin/tmp/100100_rlib_patch_donot_
modify/Oracle Interim Patch Installer version 13.1.0.0.0
Copyright (c) 2013, Oracle Corporation. All rights reserved.

Oracle Home      : /scratch/my_admin/oraclehome/fmw12/fmwhome_2
Central Inventory : /scratch/my_admin/oraInventory
   from           : /scratch/my_admin/oraclehome/fmw12/fmwhome_2/oraInst.loc
OPatch version    : 13.1.0.0.0
OUI version       : 13.1.0.0.0
Log file location : /scratch/my_admin/oraclehome/fmw12/fmwhome_
2/cfgtoollogs/patch/100100_May_09_2013_11_10_30/apply2013-05-09_11-10-21AM_1.log

OPatch detects the Middleware Home as "/scratch/my_admin/oraclehome/fmw12/fmwhome_
2"

Applying interim patch '100100' to OH '/scratch/my_admin/oraclehome/fmw12/fmwhome_
2'
Verifying environment and performing prerequisite checks...
All checks passed.
Backing up files...

Patching component oracle.fmwconfig.common.shared, 12.1.2.0.0...

There is an error with library regeneration, please refer to the log file for
details. OPatch will continue applying the patch.

Verifying the update...
Log file location: /scratch/my_admin/oraclehome/fmw12/fmwhome_
2/cfgtoollogs/patch/100100_May_09_2013_11_10_30/apply2013-05-09_11-10-21AM_1.log

OPatch failed with error code 115
```

You now need to use the OPERR utility to determine a course of action.

**4.1.1.1 Determining Whether OPERR Uses the Latest Message File**

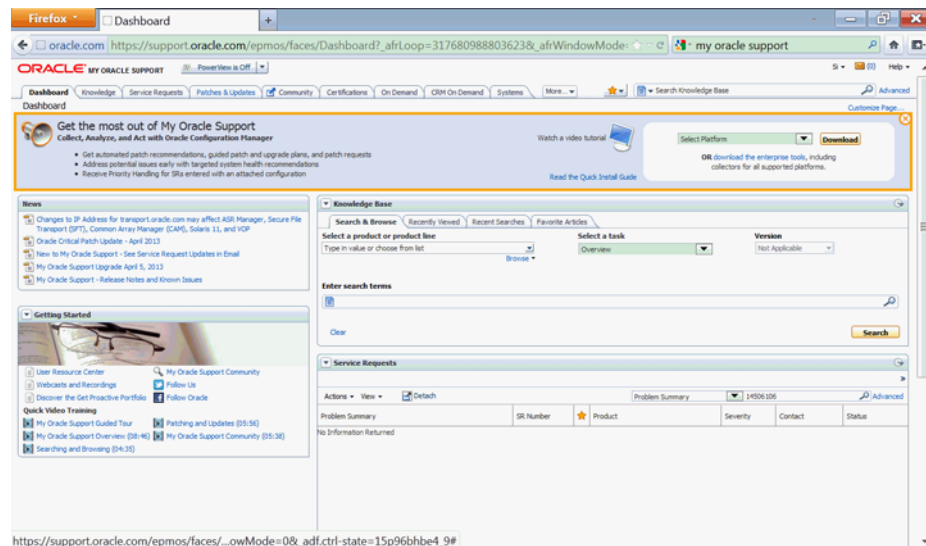
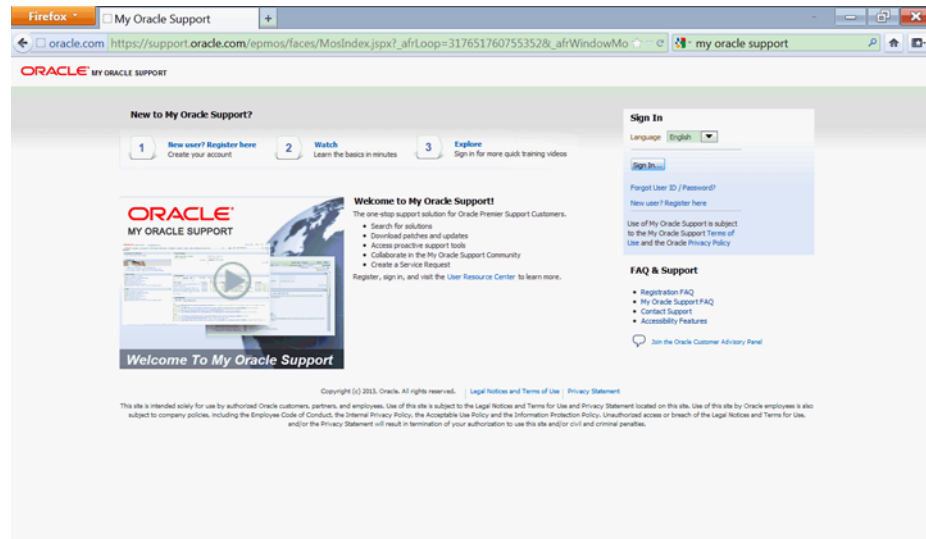
You run OPERR to obtain up-to-date troubleshooting information for error code 115, but find that the OPERR states that 115 is an undefined error code, thus indicating that the error message file used by OPERR is not the latest version.

```
bash-4.1$ OPatch/operr 115
Undefined error code.
You can get the latest message file from My Oracle Support using patch 16609471
```

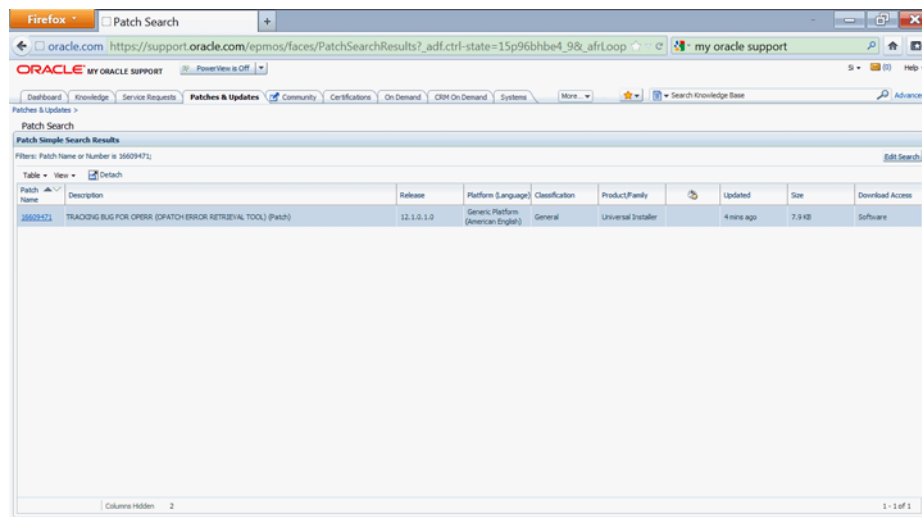
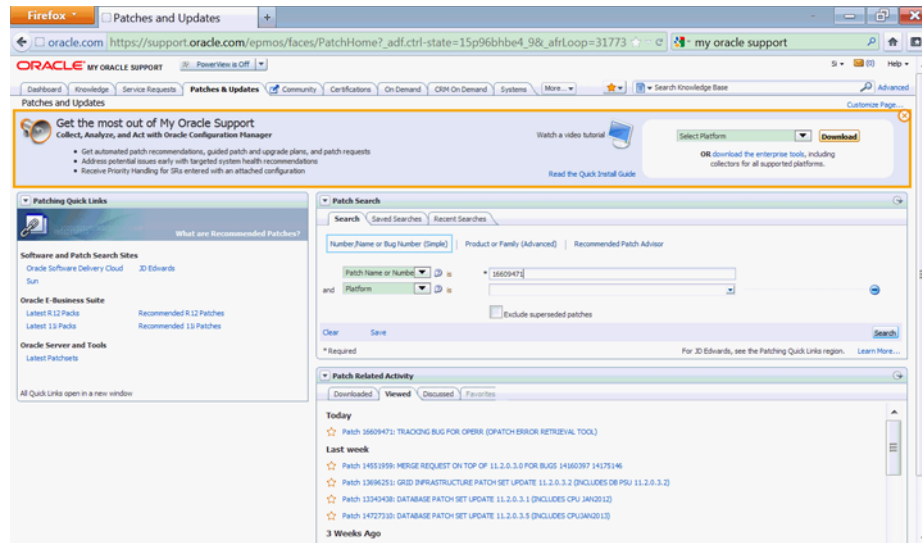
**4.1.1.2 Obtaining the Latest Message File from MOS**

The next step is to retrieve the latest message file from patch 16609471 via MOS. To access MOS, go to [support.us.oracle.com](http://support.us.oracle.com). The following sequence of graphics illustrate the steps to download the message file patch.

## Step 1: Go to <http://support.oracle.com>



[https://support.oracle.com/epmos/faces/\\_afwMode=08\\_adf.ctrl.state=15p96bhbe4\\_9#](https://support.oracle.com/epmos/faces/_afwMode=08_adf.ctrl.state=15p96bhbe4_9#)

**Step 2: Search for patch 16609471.**

### Step 3: Download the patch.



#### (Optional) Check the version of the downloaded patch.

```
bash-4.1$ OPatch/operr -version -f /scratch/my_admin/operr.txt
operr version is 12.1.0.1.1
Message file version is 12.1.0.1.1, 05/09/13
```

#### 4.1.1.3 Running OPERR with the Latest Message File

Using the -f option allows you to point OPERR to the message file you downloaded from MOS.

```
bash-4.1$ OPatch/operr 115 -f /scratch/my_admin/16609471/operr.txt
Error code: 115
Error message:OPatch failed to regenerate library.
```

## 4.2 Debugging: Enable Logging and Tracing

*Logging and tracing* is a common aid for debugging. OPatch maintains logs for all apply, rollback, and lsinventory operations. The log files are located at the following directory:

```
<ORACLE_HOME>/cfgtoollogs/patch
```

Each log file is tagged with the timestamp of the operation. Log files are named as `opatch_<date mm-dd-yyyy>_<time hh-mm-ss>.log`.

---

**Note:** A new log file is created each time OPatch is executed.

---

For example, if a log file is created on May 17th, 2013 at 11.55 PM, it will be named as follows:

```
opatch_05-17-2013_23-55-00.log
```

---

**Note:** You can set OPatch to debug mode by setting the environment variable `OPATCH_DEBUG` to `TRUE`.

---

OPatch also maintains an index of the commands executed with OPatch and the log files associated with it in the `opatch_history.txt` file located in the `<ORACLE_HOME>/cfgtoollogs/opatch` directory. A sample of the `history.txt` file is as follows:

```
Date & Time : Tue Apr 26 23:00:55 PDT 2012
Oracle Home : /private/oracle/product/11.2.0/db_1/
OPatch Ver. : 11.2.0.0.6
Current Dir : /scratch/oui/OPatch
Command    : lsinventory
Log File   :
/private/oracle/product/11.2.0/db_1/cfgtoollogs/opatch/opatch-2013_Apr_26_
23-00-55-PDT_Tue.log
```

## 4.3 References

This section to contain a list of other documentation that may provide support. The main link will point to the "Master Note for OPatch" (Doc ID 293369.1) in MOS.

Patch Set FAQ: See Patchset FAQ (Doc ID 552777.1) for a complete summary of patchset FAQs. This document is available in My Oracle Support.

Information Center: Patching and Maintaining Oracle Database Server/Client Installations (Doc ID 1351428.1 )

## 4.4 Products and Patch Types Not Supported by OPatch

Oracle now offers a wide range of products, including hardware and operating systems. However, patches required by some products are not currently supported by OPatch.

---

---

**Note:** If the Oracle product you install (such as Oracle Database or Fusion Middleware) creates an Oracle home directory, then OPatch is provided as part of that installation.

---

---

Typically, the types of Oracle products not supported by OPatch include:

- Hardware (for example, firmware updates for Sun servers).
- Operating system (for example, kernel patch updates for Oracle Linux or Oracle Solaris.)
- Java (for example, patch updates for JRE, JDK)
- Software products that do not create an Oracle home directory (for example, Oracle OpenOffice)



---

# Troubleshooting OPatchauto

This chapter describes common OPatchauto problem that may occur during usage.

This chapter covers the following:

- [OPatchauto Troubleshooting Architecture](#)
- [OPatchauto \(Use Cases\)](#)
- [Common Error Symptoms/Conditions](#)
- [OPatchauto Log Files](#)

## 5.1 OPatchauto Troubleshooting Architecture

In order for OPatchauto to fully automate the patching process, it accesses various tools/utilities to carry out different patching phases. The four primary tools/utilities are:

- **OPatch** - Applies patches to GI/RAC/DB homes.
- **rootcrs.pl** - Controls GI Home access by unlocking files so they are patchable, as well as stopping and starting instances to permit patching of GI files.
- **patchgen** - Records the patch level.
- **datapatch** - Applies SQL changes to database instances.

These tools/utilities are accessed during the patching process. Troubleshooting OPatchauto, therefore, involves diagnosing issues with the individual tools.

## 5.2 OPatchauto (Use Cases)

When using OPatchauto, problems may arise where it is not clear as to how to proceed with the resolution. The following use cases illustrate common patching scenarios where you may encounter such problems and general procedures you can use to resolve the problems.

### 5.2.1 OPatch Fails

See [Chapter 4, "Troubleshooting OPatch"](#) for more information.

### 5.2.2 Rootcrs.pl

The rootcrs.pl script performs the operations necessary to install the Oracle Clusterware stack on a node of a cluster. During an OPatchauto session, you may encounter errors stemming from the rootcrs.pl script.

### 5.2.2.1 Rootcrs.pl Prepatch

Command issued by OPatchauto: \$GRID\_HOME/crs/rootcrs.pl -prepatch

If running rootcrs.pl fails, error codes and their associated messages will be generated, as shown in the following example

```
CRS-1159: The cluster cannot be set to rolling patch mode because Oracle
Clusterware is not active on at least one remote node.
```

If the message is not clear, you can obtain additional help by running the **OERR** utility to obtain explicit cause and recommended action information.

#### Running OERR

Running OERR for a specific error code will generate both the cause and action for the specified error code. For example, running oerr for error code 1159 generates the following:

```
$GRID_HOME/bin/oerr crs 1159
```

```
Cause: The cluster could not be set to rolling patch mode because Oracle
Clusterware was not active on any of the remote nodes.
```

```
Action: Start Oracle Clusterware on at least one remote node and retry the 'crsctl
start rollingpatch' command, or retry patching using the non-rolling option.
```

The following table lists the common error codes that you may encounter during a patching session. For an exhaustive list, see the Oracle® Database Error Messages manual.

**Table 5–1 CRS Error Codes**

Error Code	Console Message
1153	There was an error setting Oracle Clusterware to rolling patch mode.
1154	There was an error setting Oracle ASM to rolling patch mode.
1156	Rejecting the rolling patch mode change because the cluster is in the middle of an upgrade.
1157	Rejecting the rolling patch mode change because the cluster was forcibly upgraded.
1158	There was an error setting the cluster to rolling patch mode.
1159	The cluster cannot be set to rolling patch mode because Oracle Clusterware is not active on at least one remote node.
1162	Rejecting rolling patch mode change because the patch level is not consistent across all nodes in the cluster. The patch level on nodes %s is not the same as the expected patch level [%u] found on nodes %s.
1163	There was an error resetting Oracle Clusterware rolling patch mode.
1164	There was an error resetting Oracle ASM rolling patch mode.
1166	Rejecting rolling patch mode change because Oracle ASM is in [%s] state.
1168	There was an error resetting the cluster rolling patch mode.

**Table 5–1 (Cont.) CRS Error Codes**

Error Code	Console Message
1171	Rejecting rolling patch mode change because the patch level is not consistent across all nodes in the cluster. The patch level on nodes %s is not the same as the patch level [%u] found on nodes %s.
1181	There was an error retrieving the Oracle Clusterware release patch level.
1183	Oracle Clusterware release patch level is [%u] and an incomplete list of patches [%s] have been applied on the local node.
1191	There was an error retrieving the Oracle Clusterware software patch level.

### 5.2.2.2 Rootcrs Problem Use Cases

#### Scenario 1: Non-rollable Patch is Applied in Rolling Mode

You have a two-node (node 1 and node 2) configuration and are attempting to apply a non-rollable patch in rolling mode.

---

**Note:** By default, OPatchauto applies patches in rolling mode.

---

Because you are applying the patch in rolling mode, you have not shut down all databases and stacks. When OPatchauto is run, it prints out the stack inventory and updates the binaries as expected.

#### Problem

When `rootcrs.pl -postpatch` (Performs the required steps after the Oracle patching tool (OPatch) is invoked) is run, it fails due to the library updates not being in sync. In this situation, OPatchauto (which runs OPatch) fails with a non-zero exit code. However, the patch is left in the GI Home. The stack cannot be brought up.

#### Recommended Action

It is important to note that, in this situation, it is not necessary to roll back the patch as it has already been successfully applied to node 1. In general, make sure any attempt to bring up the stack is the very last step performed: Even if the stack fails to come up, the patch has been successfully applied to all nodes.

Because the patch is non-rollable, to resolve the stack issue:

1. Bring down the stack on all nodes.
2. Patch the remaining nodes.
3. Bring the stack back up on all nodes.

#### Scenario 2: OPatchauto Fails to Patch the GI Home

You have a system patch containing sub-patches (P1 and P2). When OPatchauto apply is run, it will first patch the RAC homes. In this scenario P1 is applied to RAC at time t1, P2 is then applied to RAC at time t2. OPatchauto attempts to apply sub-patch P2 at time t3 to the GI Home but fails.

**Problem**

OPatchauto fails with a non-zero exit code. The error message indicates failure occurred when applying sub-patch P2 on the GI Home. Note that the error message will provide you with a log file location. The RAC Home now contains P1 and P2, but the GI Home is missing P2.

**Recommended Action**

You need to apply the missing patch to the GI Home. Because the system patch has already been successfully applied to the RAC Home, there is no need to roll back the patch.

1. From the log file, determine what caused patch application to fail for the GI Home.
2. Fix the issue that caused the GI Home patch application to fail.

When patch application fails for the GI Home, there are three possible causes:

- `patchgen --` In this situation, refer to the recommended action specified for patchgen use case. See ["Patchgen"](#) on page 5-4.

You will have to manually patch the GI Home. Refer to the patch README for instructions.

- `opatch -call` command failed. In this situation, an error occurred during OPatch execution. For example, OPatch could not copy a required file.
- `rootcrs.pl -prepatch` (perform the required steps before OPatch is invoked) fails.

Regardless of the cause of failure, you must resolve the issue and then manually patch the GI Home.

3. Re-run `opatchauto apply` on the GI Home. OPatchauto knows where patch application has been successful and where it has failed.

## 5.2.3 Patchgen

**Scenario**

When applying a system patch, OPatchauto fails as a result of error conditions encountered by patchgen.

**Problem**

OPatchauto fails with the STDOUT error message indicating a patching failure due to problems encountered by patchgen.

**Recommended Action**

1. Determine whether the error message is a result of a patchgen error. From the message output, you can determine whether or not it is of patchgen origin by searching for the keyword "patchgen." The following example shows a sample error message generated by patchgen. The keyword "patchgen" and the associated error code is in bold.

**Example 5-1 Patchgen Error Output**

```
$export ORACLE_HOME=/scratch/GI12/product/12.1.0/crs
$/scratch/GI12/product/12.1.0/crs/bin/patchgen commit -pi
13852018
loading the appropriate library for linux
```

```
java.lang.UnsatisfiedLinkError:
/scratch/GI12/product/12.1.0/crs/lib/libpatchgensh12.so (libasmclntsh12.so:
cannot open shared object file: No such file or directory)
```

2. With the patchgen error code, run the `oerr` command to obtain the cause and recommended action(s) to resolve the specific problem encountered by patchgen. Implement the suggested action. See ["Running OERR"](#) on page 5-2.
3. When patchgen errors out, it will ask whether or not you want to keep the patch or roll it back. By default, patchgen rolls back the patch. Whether or not the patch is rolled back determines your course of action in the next step.
  - If the patch was not rolled back, run `patchgen` again.  
Despite the error, the patch itself still exists in the GI/RAC/DB home since it was not rolled back.
  - If the patch has been rolled back, you may find that the OPatchauto has applied the system patch to the RAC Home, but not all sub-patches to the GI Home. At this point, you need to apply only part of the system patch to the GI Home.  
OPatch will tell you via `lsinventory`, which patches have not been applied. In order to apply specific sub-patches, you must resort to manual patching:
    1. Shut down the stack.
    2. Run `opatch apply` (not `OPatchauto`) on the GI Home.
 Refer to the patch README for explicit instructions on applying a patch manually.

The following table lists possible patchgen error codes.

**Table 5–2 Patchgen Error Codes**

Error Code	Reason	Debugging Information
2	Internal Error	Generic failure error code.
3	Internal Error	MS Windows: Resource file read error.
4	Internal Error	MS Windows: Resource file write failed.
5	Internal Error	Unix: Open for patch repository failed.
6	Internal Error	Unix: Normalization of full path libasmclntsh failed.
7	Internal Error	Unix: Write to patch repository failed.
18	Internal Error	PGA initialization failed.
19	Internal Error	Patch iterator init failed.
40	Syntax Errors, appropriate message would be displayed.	No argument to patchgen. Example: <code>\$!patchgen</code>

**Table 5–2 (Cont.) Patchgen Error Codes**

Error Code	Reason	Debugging Information
41	Syntax Errors, appropriate message would be displayed.	No arguments to patchgen commit/recover Example: patchgen commit
42	Syntax Errors, appropriate message would be displayed.	-pi patchids are not numbers. Example: patchgen commit -pi 123d
43	Syntax Errors, appropriate message would be displayed.	-rb patchids are not numbers. Example: patchgen commit -rb 123d
44	Syntax Errors, appropriate message would be displayed.	Argument to patchgen commit/recover is something other than -pi or -rb Example: patchgen recover -random
45	Syntax Errors, appropriate message would be displayed.	Patchgen invoked with invalid argument. Example: patchgen comit -pi
46	Loading libpatchgensh12.so failed.	

## 5.2.4 Datapatch

### Scenario

You attempt to run OPatchauto to patch four GI/RAC/DB homes. This patch contains both bits and SQL to update the database. When you run OPatchauto, it performs two actions:

- Applies bits to the GI/RAC/DB home
- Runs SQL (via the datapatch command)

Typically, you run OPatchauto on each GI/RAC/DB home. With each run, OPatchauto calls datapatch to run the patch SQL. datapatch will do nothing on the first three nodes (no-op). On the last (fourth) node, datapatch tries to execute the patch SQL.

If datapatch fails, you will see an error message. To find out if the error is from datapatch, view OPatchauto debug log. In the debug log there is a section for datapatch command execution that will have the diagnostic information.

### Problem

You see an error message indicating OPatchauto has failed. This can be misleading as the origin of the message is unclear. The error message was actually generated by datapatch when it failed to apply the SQL to the last node.

### Recommended Action

In general, you can ignore the error message and then run datapatch manually on the last node. After running datapatch manually, run OPatchauto again.

**Rollable VS. Non-Rollable Patches:** Patches are designed to be applied in either rolling mode or non-rolling mode. For more information about rolling versus non-rolling, see ["Patch Application Modes"](#) on page 3-2. Depending on whether the patch is rollable or non-rollable determines the course of action.

If a patch is rollable, the patch has no dependency on the SQL script. The database can be brought up without issue. Note that a rollable patch can be applied in either rolling or non-rolling mode.

If, however, the patch is non-rollable, then the patch must first be rolled back. Note that OPatchauto will prevent you from applying a non-rollable patch in rolling mode.

### Sequence

1. OPatchauto fails. Determine whether the failure is, in reality, caused by datapatch.
2. For rollable patches:
  1. Ignore data patch errors on node 1 - node(n-1).
  2. On the last node (node n), run data sync again. You can cut and paste this command from the log file.
  3. If you still encounter datapatch errors on the last node, call Oracle Support or open a Service Request.
3. For non-rollable patches:
  1. Bring down all databases and stacks manually for all nodes.
  2. Run opatchauto apply on every node.
  3. Bring up the stack and databases. Note that the databases must be up in order for datapatch to connect and apply the SQL.
  4. Manually run datapatch on each node. Note that if you do not run datapatch, the SQL for the patch will not be applied and you will not benefit from the bug fix. In addition, you may encounter incorrect system behavior depending on the changes the SQL is intended to implement.
  5. If datapatch continues to fail, you must roll back the patch. Call Oracle Support for assistance or open a Service Request.

## 5.3 Common Error Symptoms/Conditions

### 5.3.1 OPatch Apply ( OPERR Error)

See ["Error Messages"](#) on page 4-1 for more information

### 5.3.2 Rootcrs.pl Postpatch

Patch scenario where the patch attempt fails when trying to bring up the product stack.

### 5.3.3 Patcherr

Patch scenario fails due to relink failure.

## 5.4 OPatchauto Log Files

Log files allow you to diagnose issues if problems occur during your patching session. OPatchauto generates log files for both the apply and analyze operations.

An OPatchauto session generates three types of log files.

- Regular log file
- Debug log file
- Steps log file

### Regular Log File

The regular log file contains primary session information such as the GI/RAC/DB homes that were patched, the patches that were applied, and any specified patch options such as rolling or non-rolling.

### Debug Log file

The debug log file is a superset of the regular log file and is intended for use by Oracle support. In addition to including the content of a regular log file, the debug log file contains detailed session information. For example, for each command that is run the command output is also logged.

### Steps Log file

The steps log file contains only the steps that OPatchauto has generated during the patching session. This file allows you to view the steps OPatchauto has executed without having to wade through extraneous information. If problems occur during the patching session, you can see at which step patching process failed.

### 5.4.1 Log File Naming Conventions

OPatchauto log file names are generated using follow naming conventions:

**Table 5–3 Log File Naming Conventions**

Log File	Naming Convention
Regular	opatchauto<timestamp>.deploy.log
Debug	opatchauto<timestamp>.deploy.debug.log
Steps	opatchauto<timestamp>.deploy.steps.log

### 5.4.2 Log File Locations

When running OPatchauto, the screen output will display the exact location for the log files. By default, the log files can be found at the following location:

```
$GRID_HOME/cfgtoollogs/opatchauto/<PATCH_ID>/
```



## OPatch Syntax and Commands

This appendix provides a summary of the syntax and command options to use for the `opatch` command. Use these command options to develop your own patch plan.

### A.1 OPatch Syntax

The OPatch utility is located in the `$Oracle_Home/OPatch` directory. You can run it with various commands and options. The following string shows the syntax for the OPatch utility:

```
<Path_to_OPatch>/opatch [-help] [-r[eport]] [command] [-option]
```

where:

- `[-help]` displays the help message for the `opatch` command.
- `[-report]` prints the actions without executing.
- `[command]` is one of the OPatch commands described in [Table A-1](#).
- `[-option]` is one of the OPatch command options. See each command listed below for a summary of available options.

**Table A-1** *OPatch Commands*

Command	Description
<a href="#">apply</a>	Installs or applies a patch.
<a href="#">compare</a>	Compare two files generated by the <code>opatch lsinventory -xml</code> command.
<a href="#">lsinventory</a>	Lists what is currently installed on the system.
<a href="#">lspatches</a>	Prints a summary of all installed patches.
<a href="#">napply</a>	Installs <i>n</i> number of patches (hence <code>napply</code> ).
<a href="#">nrollback</a>	Rolls back patches from several GI/RAC/DB homes at the same time.
<a href="#">query</a>	Queries a given patch for specific details.
<a href="#">rollback</a>	Removes a patch.
<a href="#">version</a>	Prints the current version of the patch tool.
<a href="#">prereq</a>	Runs patching prerequisite checks on an ORACLE_HOME..
<a href="#">util</a>	Invokes specified utilities on an GI/RAC/DB home

To view additional information for any command, use the following command:

```
<Path_to_OPatch>/opatch command -help
```

If using Perl, use the following command:

```
perl opatch.pl command -help
```

To show the full syntax of the -help option, enter `opatch -h` to view the following display:

```
Usage: opatch [ -help ] [ -report ] [ command ]
```

```

        command := apply
compare
        lsinventory
        lspatches
        napply
        nrollback
        rollback
        query
        version
        prereq
        util

```

```

<global_arguments> := -help      Displays the help message for the command.
                    -report      Print the actions without executing.

```

example:

```

'opatch -help'
'opatch -help -fmw'
'opatch apply -help'
'opatch compare -help'
'opatch lsinventory -help'
'opatch lspatches -help'
'opatch napply -help'
'opatch nrollback -help'
'opatch rollback -help'
'opatch prereq -help'
'opatch util -help'

```

## A.2 apply

The `apply` command applies an interim patch to an `ORACLE_HOME` from the current directory. The patch location can be specified using the parameter `patch_location`. This command does not support System Patch.

### Syntax

Use following syntax for this command:

```

opatch apply [-connectString <List of connect strings>]
             [-delay <value> ] [ -force ] [ -force_conflict ]
             [-init <parameters for the init script in escaped double
             quotes> [-opatch_init_end] ]
             [-invPtrLoc <Path to oraInst.loc> ]
             [-jre <LOC> ] [-local ] [-local_node <Local node name>]
             [-minimize_downtime ] [-no_bug_superset ] [-no_inventory ]
             [-no_relink] [-no_sysmod] [-ocmrf <Response file location> ]
             [-oh <ORACLE_HOME> ]
             [-post <parameters for the post script in escaped
             quotes> [-opatch_post_end] ]

```

```

[-pre <parameters for the pre script
  in escaped double quotes> [-opatch_pre_end] ]
[-profile_mask <Name of profile>]
[-property_file <Path to property file>]
[-ptlConnect <portal connect string>]
[-ptlPassword <portal password>]
[-ptlSchema <portal schema>]
[-remote_nodes <List of remote nodes (node1,node2)>]
[-retry <value >] [-runSql <value>]
[silent <value>] [-sqlScript <path of the sql file>] [-verbose <value>]
[ <Patch Location> ]

```

## Options

Table A-2 describes the options available for the apply command.

**Table A-2** *apply Command Options*

Option	Description
delay	Specifies how many seconds to wait before attempting to lock the inventory in the case of a previous failure. You can use this option only if you specify the <code>-retry</code> option.
force	Removes conflicting patches from the system. If a conflict exists that prevents the patch from being applied, you can use this option to apply the patch. OPatch removes all the conflicting patches before applying the current patch.
force_conflict	If a conflict exists which prevents the patch from being applied, the <code>-force_conflict</code> flag can be used to apply the patch. OPatch will remove all the conflicting patches before applying the current patch. This will override the 'silent' behavior for conflicts and hence is meaningful only when used with the 'silent' option.
invPtrLoc	Specifies the location of the <code>oraInst.loc</code> file. This option is needed when the <code>-invPtrLoc</code> argument was used during installation. Oracle recommends using the default Central Inventory for a platform.
jre	Instructs OPatch to use JRE (Java) from the specified location instead of the default location under the GI/RAC/DB home directory.
local	Specifies that the OPatch utility should patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.  You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this argument for non-rolling patches.
local_node	Tells OPatch the local node for this cluster. You can use this option on Oracle Real Application Clusters environments.
minimize_downtime	Specifies the order of nodes that OPatch should patch.  This option only applies to Oracle Real Application Clusters environments. You cannot use it with the <code>-local</code> option or a rolling patch.
no_bug_superset	Specifies to error out if the current patch bugs-to-fix is a superset (or same set) as an installed patch bugs-fixed in the GI/RAC/DB home directory.

**Table A–2 (Cont.) apply Command Options**

Option	Description
no_relink	This option does not perform any make operations. You can use it during multiple patch applications and to perform the linking step only once. OPatch does not keep track of the make operations it did not perform. You need to make sure to execute OPatch without this option at the end for compilation.
ocmrf	Give OPatch the absolute path to the OCM response file to be used for OCM configuration. -silent must be used in conjunction with -ocmrf if GI/RAC/DB home doesn't have OCM installed and configured.
oh	Specifies the GI/RAC/DB home directory to use instead of the default. This takes precedence over the environment variable ORACLE_HOME.
opatch_post_end	Marks the end of the post option. You use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. You use this option with the pre option. If you do not use this argument, everything after pre until the end of the command is passed into pre.
post	Specifies the parameters to be passed to the post script. This script is executed after the patch is applied. You need to enclose the values for this option in double-quotes.
pre	Specifies the parameters to be passed to the pre script. This script is executed before the patch is applied. You need to enclose the values for this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
report	Prints the action to the screen without executing it.
retry	Tells OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any answers to "yes."
verbose	Prints additional OPatch output to the screen as well as to the log file.

---



---

**Note:** If a patch consists of SQL changes, follow the instructions in the patch README, which is included with the patch to apply the SQL scripts.

---



---

## A.3 compare

The `compare` command allows you to compare the bugs that have been fixed between two GI/RAC/DB homes. This command allows for comparison between two files generated by the `opatch lsinventory -xml` command. Currently, this command only accepts two files as input.

### Syntax

```
opatch compare [<file1> <file2>]
```

## A.4 lsinventory

The `lsinventory` command lists the inventory for a particular GI/RAC/DB home, or displays all installations that can be found. This command does not have any required options.

### Syntax

Use the following syntax for this command:

```
opatch lsinventory [-all ] [-all_nodes] [-bugs_fixed <asc | desc> ]
                  [-delay <value> ] [-detail ] [-group_by_date ]
                  [-inactive]
                  [-invPtrLoc <Path to oraInst.loc> ]
                  [-jre <LOC> ] [-local ]
                  [-oh <ORACLE_HOME> ] [-patch <asc | desc> ]
                  [-patch_id <asc | desc> ]
                  [-ptlConnect <portal connect string> ]
                  [-ptlPassword <portal password> ]
                  [-ptlSchema <portal schema> ]
                  [-property_file <path to property file>]
                  [-retry <value> ] [-translation_patch ]
                  [-xml <xmlFile>]
```

The following sections provide examples for the `detail`, `bugs_fixed`, and `patch desc` options. See [Table A–3](#) for descriptions of the command options.

### Options

[Table A–3](#) describes the options available for the `lsinventory` command.

**Table A–3** *lsinventory Command Options*

Option	Description
all	Reports the name and installation directory for each GI/RAC/DB home directory found.
bugs_fixed	<p>Reports bugs fixed by installed patches in a tabular format. Besides the bugs fixed, the report also displays the installed patches, installed times, and bug descriptions.</p> <p>The fixed bugs are sorted per installed patch. Default display is patches in descending order based on installed time and ascending order of bugs within each patch. You can use 'asc' (or) 'desc' with this option to enforce sort order on bugs within each patch.</p> <p>You can use this option with the <code>patch</code> or <code>patch_id</code> option to obtain sort orders with installed patches.</p>
delay	If you specify <code>retry</code> , this option tells OPatch how many seconds it should wait before attempting to lock the inventory again in case of a previous failure.
detail	Reports the installed products and other details. You cannot use this option with the <code>-all</code> option.
group_by_date	Specifies that OPatch should group all installed patches by the date they were installed in the GI/RAC/DB home.
invPtrLoc	Specifies the location of the <code>oraInst.loc</code> file. You need this option if you used the <code>invPtrLoc</code> option during the installation. Oracle recommends using the default Central Inventory for a platform.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the GI/RAC/DB home directory.

**Table A–3 (Cont.) *lsinventory* Command Options**

Option	Description
local	Instructs OPatch to display inventory information of the local node only.
oh	Specifies the GI/RAC/DB home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
patch	Lists the patch IDs installed in the GI/RAC/DB home in ascending (asc) or descending (desc) order, which is the default, based on installed time.
patch_id	Lists the patch IDs installed in the GI/RAC/DB home in ascending (asc) or descending (desc) order based on patch numbers. The value defaults to ascending (asc).
property_file	Indicates the user-defined property file that OPatch should use. The path to the property should be absolute. This property file takes precedence over the property file that OPatch supplies.
retry	Specifies how many times OPatch should retry when there is an inventory lock failure.
xml	Generates xml output based on the current GI/RAC/DB home inventory to the specified xml file.

## A.5 lspatches

The `lspatches` command prints a summary of all installed patches.

### Syntax

Use the following syntax for the `lspatches` command:

```
opatch lspatches [-bugs] [-id <patch ID> ]
                 [-invPtrLoc <Path to oraInst.loc> ] [-jre <LOC> ]
                 [-oh <ORACLE_HOME> ] [ <patch location> ]
                 [-qfile <file path> ] [-required ] [-verify]
```

### Options

[Table A–4](#) lists the options available for the `lspatches` command.

**Table A–4 *lspatches* Command Options**

Option	Description
bugs	Prints out bugs in addition to the summary
id	This option specifies the patch number. It must be registered in the GI/RAC/DB home inventory. It can be any numeric sequence or combined with language. Example: 11111, 11111/zh_CN. It cannot support multiple patch IDs.
invPtrLoc	Used to locate the oraInst.loc file. When the installation uses the invPtrLoc flag, the value should indicate the path to oraInst.loc file
jre	This option tells OPatch to use JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
oh	The GI/RAC/DB home to work on. This takes precedence over the environment variable ORACLE_HOME.

**Table A-4 (Cont.) lspatches Command Options**

Option	Description
qfile	Specifies the relative path to GI/RAC/DB home of the file to determine the latest patch that touches this file. Example:  On Linux: admin/rdbms/catcpu.sql On Windows: admin\rdbms\catcpu.sql  OPatch can tell which latest patch touches the file catcpu.sql in the GI/RAC/DB home.
required	This option will print key metadata only. This includes the following metadata: required components, prereq patches, executables to shutdown and support platforms.  This option should be accompanied by either option -id <PATCH_ID> or <PATCH_LOCATION>.
verify	This option verifies whether or not the specified patch ID or patch location is registered in the GI/RAC/DB home inventory. In addition, this option validates all patch files in the GI/RAC/DB home.  This option should be accompanied by either option -id <PATCH_ID> or <PATCH_LOCATION>. This option doesn't support System Patch.  Example: opatch lspatches -id 111 -verify opatch lspatches /scratch/test/111 -verify

## A.6 napply

This command applies patches to several GI/RAC/DB homes at the same time. This command does not support System Patches.

### Syntax

Use the following syntax for the napply command:

```
opatch napply [patch_location] [-id comma-separated list of patch IDs]
    [ -all_nodes ]
    [-connectString <List of connect strings>]
    [-delay <value> ] [ -force ] [ -force_conflict ]
    [-idFile <path of the file that has list of patch IDs> ]
    [-init <parameters for the init script in escaped double
        quotes> [-opatch_init_end] ]
    [-invPtrLoc <Path to oraInst.loc> ]
    [-jre <LOC> ] [ -local ]
    [ -local_node <Local node name> ]
    [-minimize_downtime ] [-no_bug_superset ]
    [-no_inventory ] [-no_relink]
    [-no_sysmod] [-ocmrf <Response file location> ]
    [-oh <ORACLE_HOME> ]
    [ -phBaseDir <Path to the directory that contains list
        of patch directories> ]
    [ -phBaseFile <Path to the file containing the
        location of the patches to be applied> ]
    [-post <parameters for the post script in
        escaped double quotes> [-opatch_post_end] ]
    [-pre <parameters for the pre script in
        escaped double quotes> [-opatch_pre_end] ]
    [-profile_mask <Name of profile>]
    [ -property_file <Path to property file> ]
    [-ptlConnect <portal connect string>]
    [-ptlPassword <portal password>]
```

```
[-ptlSchema <portal schema>]
[ -remote_nodes <List of remote nodes (node1,node2)> ]
[-retry <value> ] [-runSql] [-silent ]
[-skip_subset]
[-skip_duplicate]
[-sqlScript <path of the sql file>]
[-verbose ]
```

## Options

Table A-5 lists the options available for this command.

**Table A-5** *napply Command Options*

Option	Description
delay	Specifies how many seconds to wait before attempting to lock the inventory again for a previous failure. You can use this option only if you specify the <code>retry</code> option.
force	Removes conflicting patches from the system. If a conflict exists that prevents the patch from being applied, you can use this option to apply the patch. OPatch removes all the conflicting patches before applying the current patch.
force_conflict	If a conflict exist which prevents the patch from being applied, the <code>-force_conflict</code> flag can be used to apply the patch. OPatch will remove all the conflicting patches before applying the current patch. This will override the 'silent' behavior for conflicts and hence is meaningful only when used with 'silent' option.
id	Use the 'lsinventory' option to display all patch ids. Each one-off patch is indicated by its id. A comma separated list of patches can be given to select the patches to be applied. For translation patches, the patch id should be of the format <Patch ID>/<Language code>.
idFile	The input to be given is a file location that contains a list of apply patch ids separated by commas or white spaces. This option cannot to be in conjunction with 'id' option. For translation patches, the patch id should be of the format <Patch ID>/<Language code>.
invPtrLoc	Specifies the location of the <code>oraInst.loc</code> file. The <code>invPtrLoc</code> option is needed when this option is used during installation. Oracle recommends the use of the default Central Inventory for a platform.
jre	Instructs OPatch to use JRE (Java) from the specified location instead of the default location under the GI/RAC/DB home directory. You cannot specify the <code>jdk</code> and <code>jre</code> options together.
local	<p>Specifies that OPatch should patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.</p> <p>You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.</p>
no_bug_superset	Specifies to error out if the current patch's bugs-to-fix is a superset (or same set) of an installed patch's bugs-fixed in the GI/RAC/DB home directory.



**Table A-5 (Cont.) napply Command Options**

Option	Description
no_relink	This option does not perform any make operations. You can use it during multiple patch applications and to perform the linking step only once. OPatch does not keep track of the make operations it did not perform. You need to make sure to execute OPatch without this option at the end for compilation.
ocmrf	Give OPatch the absolute path to the OCM response file to be used for OCM configuration. -silent must be used in conjunction with -ocmrf if GI/RAC/DB home does not have OCM installed and configured.
oh	Specifies the GI/RAC/DB home directory to use instead of the default. This takes precedence over the environment variable ORACLE_HOME.
opatch_post_end	Marks the end of the post option. You use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. You use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
phBaseDir	Used to specify a directory containing patch directories (or) zip files.
phBaseFile	If you do not specify <patch_location>, use this option to point OPatch to a file containing a list of patches to be n-applied. Each line in the file points to a location of a patch.
post	Specifies the parameters to be passed to the post script. This script is executed after the patch is applied. You need to enclose the values for this option in double-quotes.
pre	Specifies the parameters to be passed to the pre script. This script is executed before the patch is applied. You need to enclose the values for this option in double-quotes.
profile_mask	If the patch to be applied specifies WLS patch/patchset as prerequisites, OPatch will read the WLS default patch profile. To have OPatch read non-default patch profile, specify the patch profile name with this option.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
report	Prints the action to the screen without executing it.
retry	Tells OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any answers to "yes."
skip_duplicate	Skips patches to be applied that are duplicates of other patches installed in the GI/RAC/DB home. Two patches are duplicates if they fix the same set of bugs.
skip_subset	<p>Skips patches to be applied that are subsets of other patches installed in the GI/RAC/DB home. One patch is a subset of another patch if the former fixes a subset of bugs fixed by the latter.</p> <p>For example, if you used napply yesterday for patch A that fixed bugs 1 and 2, you use napply today with this option for patch B that fixes bug 1 and patch C that fixes bugs 1, 2, and 3. Then subset patch A is skipped, and patch C then becomes a superset of patch A.</p>

**Table A–5 (Cont.) *napply* Command Options**

Option	Description
verbose	Prints additional OPatch output to the screen as well as to the log file.

**Examples:**

```
'opatch napply <patch_location>' to apply all patches under
<patch_location> directory
```

```
'opatch napply <patch_location> -id 1,2,3' to apply patches
1, 2, and 3 which are present under <patch_location>
directory
```

```
'opatch napply <patch_location> -skip_subset -skip_duplicate'
to apply all patches under <patch_location> directory.
OPatch will skip duplicate patches and subset patches
(patches under <patch_location> that are subsets of patches
installed in the Oracle Home)
```

```
'opatch napply <patch_location> -id 1,2,3 -skip_subset -skip_duplicate'
to apply patches 1, 2, and 3 which are under <patch_location>
directory. OPatch will skip duplicate patches and subset patches
(patches under <patch_location> that are subsets of patches
installed in the Oracle Home)
```

```
'opatch napply <patch_location> -idfile /tmp/list.txt' where list.txt contains
a list of patch IDs to be applied. The list should be separated by a space or
comma. For example: 1 2 3
```

```
'opatch napply <patch_location> -id 1/fr,2/de' to apply patches 1 (french
patch), 1 (german patch) which are present in the <patch_location> directory
```

## A.7 nrollback

The Nrollback command rolls back patches from several GI/RAC/DB homes at the same time.

### Syntax

Use the following syntax for this command:

```
opatch nrollback -id <comma-separated list of patch IDs>
[ -all_nodes ]
[-connectString <List of connect strings>]
[-delay <value>] -id <Comma separated list of patch IDs>
[-idFile <file location containing a list of
rollback IDs separated by commas or white spaces>
[-init <parameters for the init script in escaped double
quotes> [-opatch_init_end] ]
[-invPtrLoc <Path to oraInst.loc> ]
[-jre <LOC> ] [-local]
[-local_node <Local node name>]
[-no_inventory] [-no_relink] [-no_sysmod]
[-oh <ORACLE_HOME> ]
[-post <parameters for the post script in
escaped double quotes> [-opatch_post_end] ]
[-pre <parameters for the pre script in
escaped double quotes> [-opatch_pre_end] ]
```

```

        [-property_file <Path to property file>]
[-ptlConnect <portal connect string>]
[-ptlPassword <portal password>]
[-ptlSchema <portal schema>]
        [-remote_nodes <List of remote nodes (node1,node2)>]
        [-retry <value>] [-runSql] [-silent]
        [-sqlScript <path of the sql file>]
        [-verbose]

```

## Options

Table A-6 lists the options available for this command.

**Table A-6 nrollback Command Options**

Option	Description
delay	If you use the <code>retry</code> option with the <code>rollback</code> command, specifies how many seconds OPatch should wait before attempting to lock the inventory again if a previous failure occurs.
id	Indicates the patch to be rolled back. Use the <code>lsinventory</code> option to display all patch identifiers. Each one-off patch is indicated by its ID. To successfully roll back a patch, you must provide the patch identifier.
idFile	Use 'lsinventory' option to display all patch ids. Each one-off patch is indicated by its id. To rollback a patch, the id for that patch must be supplied. The input to be given is a file location that contains a list of rollback patch ids separated by commas or white spaces. For translation patches, the patch id should be of the format <Patch ID>/<Language code>. This option cannot to be in conjunction with 'id' option.
invPtrLoc	Specifies the location of the <code>oraInst.loc</code> file. You need to use this option if you used the <code>invPtrLoc</code> option during installation. Oracle recommends the use of the default Central Inventory for a platform.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the GI/RAC/DB home directory.
local	Specifies that OPatch roll back the local node, then updates the inventory of the local node. It does not propagate the patch or inventory update to other nodes.  You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
no_relink	This option does not perform any <code>make</code> operation in the patch. You can use this option during multiple patch removals and to perform the compilation step only once.
oh	Specifies the GI/RAC/DB home directory to use instead of the default directory. This takes precedence over the <code>ORACLE_HOME</code> environment variable.
opatch_post_end	Marks the end of the <code>post</code> options. Use this option with the <code>post</code> option. If you do not use this option, everything after <code>post</code> until the end of the command is passed into <code>post</code> .
opatch_pre_end	Marks the end of the <code>pre</code> options. Use this option with the <code>pre</code> option. If you do not use this option, everything after <code>pre</code> until the end of the command is passed into <code>pre</code> .

**Table A-6 (Cont.) nrollback Command Options**

Option	Description
post	This option is used to pass parameters to the post script. This script is executed after removal of the patch. The value for this option have to be enclosed in double quotes. The parameters will be common parameters which will be passed to post scripts of all patches being rolled back. This option should be ended by option 'opatch_post_end'.
pre	This option is used to pass parameters to the pre script. This script is executed before removal of the patch. The value for this option have to be enclosed in double quotes. The parameters will be common parameters which will be passed to pre scripts of all patches being rolled back. This option should be ended by option 'opatch_pre_end'.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
report	Prints the actions to the screen without executing them.
retry	Instructs OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any yes/no questions to "yes". An Oracle Real Application Clusters setup does not support this option.
verbose	Prints additional OPatch output to the screen as well as to the log file.

**Examples**

```
'opatch nrollback -id 1,2,3' to roll back patches 1, 2, and 3
that have been installed in the Oracle Home.
```

```
'opatch nrollback -id 1/fr,2/de to rollback patches 1 with language 'fr',
2 with language 'de' that have been installed in the Oracle Home.
```

**A.8 rollback**

The rollback command removes an existing one-off patch from the appropriate GI/RAC/DB home directory indicated by the reference ID. The following syntax is used for this command:

**Syntax**

```
opatch rollback -id <ID> [-connectString <List of connect strings>]
[-delay <value>]
[-init <parameters for the init script in escaped double
quotes> [-opatch_init_end] ]
[-invPtrLoc <Path to oraInst.loc> ]
[-jre <LOC> ] [-local]
[-local_node <Local node name>] [-no_inventory]
[-no_relink] [-no_sysmod]
[-oh <ORACLE_HOME>] [-ph <Patch Location>]
[-post <parameters for the post script in escaped
double quotes> [-opatch_post_end] ]
[-pre <parameters for the pre
script in escaped double quotes> [-opatch_pre_end] ]
[-property_file <path to property file>]
[-ptlConnect <portal connect string>]
```

```

        [-ptlPassword <portal password>]
[-ptlSchema <portal schema>]
        [-remote_nodes <List of remote nodes (node1,node2)>]
[-retry <value>] [-runSql] [-silent]
[-sqlScript <path of the sql file>] [-verbose]
[all_subpatches]

```

## Options

Table A-7 describes the options available for the rollback command.

**Table A-7** *rollback Command Options*

Option	Description
delay	If you use the retry option with the rollback command, specifies how many seconds OPatch should wait before attempting to lock the inventory again if a previous failure occurs.
id	Indicates the patch to be rolled back. Use the lsinventory option to display all patch identifiers. Each one-off patch is indicated by its ID. To successfully roll back a patch, you must provide the patch identifier.
invPtrLoc	Specifies the location of the oraInst.loc file. You need to use this option if you used the invPtrLoc option during installation. Oracle recommends the use of the default Central Inventory for a platform.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the GI/RAC/DB home directory.
local	Specifies that OPatch roll back the local node, then updates the inventory of the local node. It does not propagate the patch or inventory update to other nodes.  You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
local_node	Specifies to OPatch that this is the local node for the cluster to be used for rollback.  You can use this option for Oracle Real Application Clusters environments.
no_relink	Do not perform the make operations in the patch. This option can be used during multiple patch removals and perform the compilation step only once.
oh	Specifies the GI/RAC/DB home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
opatch_post_end	Marks the end of the post options. Use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. Use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
ph	Specifies the valid patch directory area. Rollback uses the command types found in the patch directory to identify which commands are used for the current operating system.
post	Specifies the parameters to be passed inside the post script. This script executes after the patch is removed. You must enclose the value of this option in double-quotes.

**Table A-7 (Cont.) rollback Command Options**

Option	Description
pre	Specifies the parameters to be passed inside the pre script. This script executes before the patch is removed. You must enclose the value of this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
report	Prints the actions to the screen without executing them.
retry	Instructs OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any yes   no questions to "yes". An Oracle Real Application Clusters setup does not support this option.
sqlScript	This option can be used to specify the custom SQL script to be run by OPatch after patching is completed
verbose	Prints additional OPatch output to the screen as well as to the log file.
all_subpatches	This option is valid ONLY for composite patches. It allows the user to rollback all sub-patches of a composite series in one shot.

## A.9 query

The query command queries a specific patch for specific details. It provides information about the patch and the system being patched.

### Syntax

Use the following syntax for this command:

```
opatch query [-all] [-is_auto_patch] [-is_translatable_patch]
             [-get_base_bugs] [-get_component] [-get_os] [-get_date]
             [-get_patch_language] [-get_patch_type] [-get_patching_model]
             [-get_product_family] [-has_sql] [-is_online_patch]
             [-is_rolling_patch] [-is_system_patch] [-jre <LOC> ] [-oh <LOC> ]
             [ <patch_location> ]
```

### Options

[Table A-8](#) lists the options available for the query command.

**Table A-8 query Command Options**

Option	Description
all	Retrieves all information about a patch. This is equivalent to setting all available options.
is_auto_patch	This option says 'true' if the patch is auto-enabled, 'false' otherwise. This command doesn't support System Patch.
is_system_patch	This option says 'true' if the patch is a System Patch, 'false' otherwise.
is_translatable_patch	This option says 'true' if the patch is translatable, 'false' otherwise. This option doesn't support System Patch.
get_base_bug	Retrieves bugs fixed by the patch.
get_component	Retrieves components the patch affects.

**Table A–8 (Cont.) query Command Options**

Option	Description
get_date	Retrieves the patch creation date and time.
get_os	Get platforms for which this patch could be applied. This option does not support System Patch.
get_patch_language	Get the language supported by the patch. This option doesn't support System Patch.
get_product_family	Get the product family to which the patch belongs. This option does not support System Patch.
is_online_patch	Indicates true if the patch is an online patch. Otherwise, the option is false.
is_rolling_patch	Indicates true if the patch is a rolling patch. Otherwise, the option is false.
jre	This option tells OPatch to use JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
oh	Specifies the GI/RAC/DB home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.

## A.10 version

The version command shows the current version number of the OPatch utility.

### Syntax

The following syntax is used for this command:

```
opatch version [-all] [-invPtrLoc <Path to oraInst.loc>]
               [-jre <LOC>] [-oh <ORACLE_HOME>]
               [-v2c <5-digit version> -oui_loc <Custom OUI Location>
               -ph <Patch Location> -ohs <list of Oracle Homes>
               separated by commas]
               [-help] [-h]
```

**Table A–9 version Command Options**

Option	Description
all	This option displays versions of OPatch for all GI/RAC/DB homes registered in the Central Inventory.
invPtrLoc	Used to locate the oraInst.loc file. When the installation used the invPtrLoc flag. This should be the path to the oraInst.loc file.
jre	This option tells OPatch to use JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
oh	The GI/RAC/DB home to work on. This takes precedence over the environment variable ORACLE_HOME
v2c	The standard 5-digit version to compare. If this option is specified with a valid version which made by no more than 5 numbers separated by '.', those GI/RAC/DB homes with valid version will be break up to two parts, one is those GI/RAC/DB homes which have opatch version higher or equal to the value of this option, and the other is those with lower version

**Table A–9 (Cont.) version Command Options**

Option	Description
ph	The patch location from where the Minimum OPatch Version (if defined) will be picked. If a valid patch location is provided, will take the required minimum opatch version from the patch and set it as the standard OPatch version to be compared to.
ohs	List of GI/RAC/DB homes to be considered. Please provide them separated by commas
help	Display valid options can be attached to this operation.

## A.11 prereq

This operation runs the prerequisite checks on an ORACLE\_HOME. This command does not support System Patches.

### Syntax

```
opatch prereq <command> [-id <Comma separated list of patch IDs>]
[-invPtrLoc <Path to oraInst.loc> ]
[-jre <LOC>] [-local_node <Local node name>]
[-oh <ORACLE_HOME> ]
[-ph <Path to the single patch location>]
[-phBaseDir <Path to the dir containing all patches>]
[-phBaseFile <Path to the file containing the
location of the patches to be applied>]
[-property_file <Path to property file>]
[-remote_nodes <List of remote nodes (node1,node2)>]
[-sid <Comma separated list of database SIDs>]
[-connectString <List of connect strings>]
```

### Commands

The prereq command executes commands that check for the prerequisite conditions shown in the table.

**Table A–10 prereq Commands**

Command	Description
CheckActiveFilesAndExecutables	Check if there are any file(s) that are active, which are touched by the patch to be applied or rolled back.
CheckActiveServices	Check for the services that are active. Note: Applicable for Windows platforms only.
CheckApplicable	Check for the presence of the required components in the ORACLE_HOME and check if all the actions of the given patch(es) are applicable.
CheckApplicableProduct	Check if the patch is applicable for the given GI/RAC/DB home. If the patch is marked for stand-alone homes, then it can not be applied on normal OUI-based home and vice versa. Also, a patch can be marked as a hybrid patch, where it is applicable for both homes.
CheckCentralInventoryForOH	Check if the given ORACLE_HOME is registered in the central inventory specified by the oraInst.loc file.
CheckCentralInventoryForRWSession	Check if a RW (read-write) session can be created for the given central inventory.



**Table A-10 (Cont.) prereq Commands**

<b>Command</b>	<b>Description</b>
CheckCentralInventoryLocation	Validate the Central Inventory location. Check if it has the correct directory structure and has the inventory.xml with read permissions.
CheckComponents	Check for the presence of the required components in the ORACLE_HOME.
CheckConflictAgainstOH	Check if there are any conflicts between the patch(es) to be applied and the patch(es) in the OH.
CheckConflictAgainstOHWithDetail	Check if there are any conflicts between the patch(es) to be applied and the patch(es) in the OH, by giving out the detailed information about the conflicts/supersets.
CheckConflictAmongPatches	Check if there are any conflicts among the patch(es) to be applied.
CheckConflictAmongPatchesWithDetail	Check if there are any conflicts among the patch(es) to be applied, by giving out the detailed information about the conflicts/supersets.
CheckFileVersions	Check if the copy actions of Fusion Applications patch(es) have at least one or more file version(s) greater than the version(s) installed in the GI/RAC/DB home.
CheckFusionAppsCompatible	Check if OUI for the GI/RAC/DB home supports patching of Fusion applications.
CheckForIdenticalPatchInOracleHome	Check if the given list of patch(es) are identical with respect to the patch(es) installed in the GI/RAC/DB home.
CheckForInputValues	Check if the input values provided to OPatch are enough for OPatch to proceed further.
CheckForNoOpPatches	Check if any of the patch(es) provided by the user are no-op patches. A no-op patch cannot be applied to the GI/RAC/DB home and can be skipped. This prerequisite will fail for no-op. patches.
CheckIfOHLockedForPatching	Check if the ORACLE_HOME is locked for patching by any previous unsuccessful OPatch Session.
CheckInstalledOneOffs	Check if all the patches provided by the user to rollback are present in the given GI/RAC/DB home.
CheckMinimumOPatchVersion	Check if all the patches provided by the user satisfy the requirement of minimum OPatch version for the OPatch currently being used.
CheckOneOffSuperset	Check if the given input Fusion Applications patch list are all candidates for one-off (or) singleton supersets. This prereq does not do any checks among the input patch list.
CheckOracleHome	Check if the given ORACLE_HOME is valid. Check if it has the inventory files with proper permissions.
CheckOraInstLocation	Check if the oraInst.loc file is proper and has the read permissions.
CheckOUILocation	Check the ORACLE_HOME for the presence of OUI.
CheckOUIVersionCompatible	Check if the OUI in the ORACLE_HOME is compatible for the OPatch.
CheckPatchApplicableOnCurrentPlatform	Check if the given patch(es) is applicable on the current platform.
CheckPatchApplyDependents	Check if all the patch(es) required by the patch(es) currently being installed is present in the GI/RAC/DB home or not.

**Table A-10 (Cont.) prereq Commands**

Command	Description
CheckPatchRollbackDependents	Check if there are any patch(es) in the GI/RAC/DB home that are depending on the patch(es) being currently rolled back.
CheckPatchShipHome	Check if the given patch to be applied has the proper structure and has the correct permissions for all the files.
CheckRemoteCommandInvocable	Check if commands can be invoked on the remote machines.
CheckRemoteCopyAndRemove	Check if files can be copied to and removed from the remote machines.
CheckRequiredLibs	Check if all the required OUI libraries are present in the given ORACLE_HOME.
CheckRollbackable	Check if the given patch(es) can be rolled back from the ORACLE_HOME.
CheckSystemCommandAvailable	Check if all the commands required for applying or rolling back the given patch are present in the system.
CheckSystemSpace	Check if enough system space is available for the patch(es) to be applied.
CheckUserAdminPrivilege	Check if the user is 'root'.  Note: OPatch should not be invoked by 'root', if so then this check fails.
CheckPatchingModel	Check if the patching model of all incoming patch(es) is compatible with that of the GI/RAC/DB home.

**Table A-11 prereq Options**

Option	Description
connectString	This option can be used to specify the list of database instances and remote nodes. The value for this option is specified as per the following syntax "SID:User:Passwd:Node". The SID is a must, others can be ignored, OPatch takes default values for it.  Example: oracle:dba:dba:mymachine,oracle1::  NOTE: If the system is not part of RAC setup, then to specify just the local node, provide the node name as empty string. This option cannot be used along with 'sid' option.
id	This option can be used to specify the patch IDs of all the patches that are to be rolled back from the given GI/RAC/DB home.
invPtrLoc	Used to locate the oraInst.loc file. Needed when the installation used the -invPtrLoc flag. This should be the path to the oraInst.loc file.
jre	This option tells OPatch to use JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
local_node	This option can be used to specify to OPatch the local node name to be used for RAC mode application of the patch.
oh	The GI/RAC/DB home to work on. This takes precedence over the environment variable ORACLE_HOME.
ph	This option can be used to specify the path to the patch location. Example: /tmp/101010

**Table A-11 (Cont.) prereq Options**

Option	Description
phBaseDir	This option can be used to specify the path to the base directory where all the patches to be applied are kept. Note: The directory should contain only non-duplicate patches in zipped or unzipped format.
phBaseFile	This option can be used to specify complete path to the file containing the location of the patches to be applied.
property_file	The user defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that is supplied with OPatch.
sid	This option can be used to specify the SIDs of the database instances. This option can be used only for local system operations.

## A.12 util

The `util` command invokes the chosen utilities on an `ORACLE_HOME`. This command does not support System Patches.

### Syntax

```
opatch util [ -help ] [ COMMAND ]
```

Run `opatch util [ COMMAND ] -help` to get help on a specific command.

**Table A-12 util Commands**

Option	Description
CheckMinimumOpatchVersion	Check if a patch is compatible with the given OPatch version.
CheckComponents	Check if the given patch is suitable for the GI/RAC/DB homes registered in the Central Inventory by components check.
Cleanup	Remove the backup for restore area of the given patch or for all the patches.
DisableOnlinePatch	Disable and remove the specified online patch(es) on the given database instances.
EnableOnlinePatch	Install and enable the specified online patch(es) on the given database instances.
GetPatchLevel	Return the patching level on Local Grid Home.
InstallOCM	Install and configure OCM.
LoadXML	Prompt for path/name of the XML file, then check if the XML is correct.
SaveConfigurationSnapshot	Save configuration snapshot of current GI home to specified file. <code>ORACLE_HOME</code> shall point to GI home. Default snapshot file is <code>ORACLE_HOME/cfgtoollogs/opatch/sysconfig/configData.txt</code>
UpdateOPatchVersion	Update the version of OPatch in the inventory of GI/RAC/DB home.
Verify	Using the defined <code>ORACLE_HOME</code> and given patch location via <code>-ph</code> , the program will check to make sure the patch was applied to the <code>ORACLE_HOME</code> .  Example: 'opatch util verify -ph /tmp/patchLoc'



---

## OPatchauto Syntax and Commands

---

This appendix provides a comprehensive listing and description of all OPatchauto commands.

---

**IMPORTANT:** OPatchauto commands must be run from the GI Home.

---

### B.1 OPatchauto Commands

The OPatchauto commands are run from the GI Home out of the standard OPatch directory. In the following generic example, an OPatchauto command is run from a GI Home.

```
$GI_HOME/OPatch/opatchauto apply <PATH_TO_PATCH_DIRECTORY>
```

where <PATH\_TO\_PATCH\_DIRECTORY> is the full path to local staging area where you have downloaded your patches.

OPatchauto consists of five primary command:

- [apply](#)
- [resume](#)
- [rollback](#)
- [version](#)

#### OPatchauto Help

You can view online help for any command by specifying the `-help` or `-h` option.

For example:

```
> opatchauto -help
> opatchauto -h
> opatchauto apply -help
> opatchauto resume -help
> opatchauto rollback -help
> opatchauto version -help
```

#### B.1.1 apply

Apply a System Patch to a GI/RAC/DB home. User specified the patch location or the current directory will be taken as the patch location.

**Important:** OPatchauto must be run from the GI Home as a root user.

### Syntax

```
<GI_HOME>/OPatch/opatchauto apply
    [-analyze]
    [-database <database names> ]
    [-generateSteps]
    [-invPtrLoc <Path to oraInst.loc> ]
    [-jre <LOC> ]
    [-norestart ]
    [-nonrolling ]
    [-ocmrf <OCM response file location> ]
    [-oh <ORACLE_HOME> ]
    [ <Patch Location> ]
```

### Options

The following table describes the options available for the `apply` command.

**Table B–1** `apply` Command Options

Option	Description
-analyze	The <code>analyze</code> option runs all the required prerequisite checks to confirm the patchability of the system without actually patching or affecting the system in any way.
-database	The <code>database</code> option is used to specify the RDBMS home(s) to be patched. This option takes a comma-separated list of Oracle database names as input.
-generateSteps	The <code>generate</code> option creates the manual steps for an <i>apply</i> session. OPatchauto performs these steps to apply the patch.
-invPtrLoc	The <code>invPtrLoc</code> option is used to locate the Central Inventory Pointer File ( <code>oraInst.loc</code> ) . Input for this option is the path to the <code>oraInst.loc</code> file.
-norestart	The <code>norestart</code> option instructs OPatchauto not to restart the Grid Infrastructure stack after patching.
-nonrolling	The <code>nonrolling</code> option instructs OPatchauto to run the patching session run in 'nonrolling' mode. Before the patching session can start, the following prerequisites must be met: <ul style="list-style-type: none"> <li>■ The stack on the local node must be running</li> <li>■ All remote nodes must be down.</li> </ul>
-ocmrf	The <code>ocmrf</code> option specifies the absolute path to the Oracle Configuration Manager (OCM) response file. It is required if the target GI/RAC/DB home does not have OCM installed and configured.
-oh	The <code>oh</code> option specifies the GI/RAC/DB home to be patched. This can be a RAC Home or Grid Infrastructure Home.
<patch location>	The explicit path to the location of the patch. If the patch location is not specified, then the current directory is taken as the patch location.

#### Example B–1 Patch a GI Home and all RAC Homes

```
<GI_HOME>/OPatch/opatchauto apply
```

#### Example B–2 Patch Databases Running from RAC Homes Only

```
<RAC_HOME>/OPatch/opatchauto apply -database db1,db2...dbn
```

**Example B-3 Patch Software-only Installations**

```
<RAC_HOME>/OPatch/patchauto apply -oh <RAC_HOME>
OR
<GI_HOME>/OPatch/patchauto apply -oh <GI_HOME>
```

**Table B-2 OPatchauto Command Examples**

Command	Action	Comment
<GI_HOME>/OPatch/patchauto apply <SYSTEM_PATCH_TOP_DIR> <GI_HOME>	Patch <b>both</b> GI home and all RAC homes.	OPatchauto must run from Grid Infrastructure home directory (GI HOME).
<GI_HOME>/OPatch/patchauto apply <SYSTEM_PATCH_TOP_DIR> -oh <GI_HOME>	Patch GI home <b>only</b> .	Local stack can be down. If local stack is down, patchauto should continue patching.  patchauto must run from the GI HOME.
<RAC_HOME>/OPatch/patchauto apply <SYSTEM_PATCH_TOP_DIR> -database db1,db2	Patch database db1 and db2 and their associated GI/RAC/DB homes.	db1 and db2 can run from the same or different GI/RAC/DB homes.  Stack has to be up. If not, error out.  If there is another db-99 running from the same GI/RAC/DB home as db1, patchauto will error out in both -silent and non-silent mode telling users that db-99 will get impacted. Users have to specify all databases or use the -oh option.  patchauto <i>must</i> run from any of the RAC homes being patched.
<RAC_HOME>/OPatch/patchauto apply <SYSTEM_PATCH_TOP_DIR> -oh <RAC_HOME>	Patch all targets running out from the given RAC home.	patchauto must run from the <RAC_HOME> being patched.  If there are many RAC homes to patch, just repeat this command many times.  Use the -oh option to patch software-only installed homes or RAC homes where database instances are not created (or created by not registered with OCR).

**Notes:**

If patchauto apply is run and encounters an individual patch within a patch set that cannot be installed, that patch will be skipped and OPatchauto will continue with the installation of the next patch in the sequence.

If patchauto apply is run and encounters an individual patch that is identical (same patch ID and Unique Patch Identifier (UPI)) to a patch already installed in the GI/RAC/DB home, OPatchauto perform the following based on specific patch conditions:

- If the individual patch was created later than the GI/RAC/DB home patch, OPatchauto installs the individual patch.
- If the individual patch's creation date is the same as the GI/RAC/DB home patch, OPatchauto will skip installing the individual patch.
- If the individual patch was created before the GI/RAC/DB home patch, an error will be generated.

This `analyze` option simulates an `opatchauto apply` session by running all prerequisite checks, when possible, without making changes to the system (either bits or configurations). Because the `analyze` command does not modify the system, it will perform the following checks:

- Run SQL sync in `analyze` mode.
- Validate all pre and post processing steps making sure the command is present and executable.

## B.1.2 resume

Resume the last OPatchauto session. Currently, it this command can only be used for resuming from Reboot Patching.

**Important:** OPatchauto must be run from the GI Home as a root user.

### Syntax

```
<GI_HOME>/OPatch/opatchauto resume [-invPtrLoc <Path to oraInst.loc>]
                                     [-jre <LOC>]
                                     [-oh <ORACLE_HOME>]
                                     [-reboot]
```

### Options

The following table describes the options available for the `resume` command.

**Table B–3** `resume` *Command Options*

Option	Description
-invPtrLoc	The <code>invPtrLoc</code> option is used to locate the Central Inventory Pointer File ( <code>oraInst.loc</code> ) . Input for this option is the path to the <code>oraInst.loc</code> file.
-jre	This <code>jre</code> option instructs OPatchauto to use the JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
-oh	The <code>oh</code> option specifies the GI/RAC/DB home to work on. This takes precedence over the environment variable <code>ORACLE_HOME</code> .
-reboot	This option tells <code>opatchauto</code> to resume from the previous halted reboot patching session.

## B.1.3 rollback

Roll back the applied System Patch from a GI/RAC/DB home. You must specify the patch location or the current directory will be used as the patch location.

**Important:** OPatchauto must be run from the GI Home as a root user.

### Syntax

```
<GI_HOME>/OPatch/opatchauto rollback
                                     [-analyze]
                                     [-database <database names> ]
                                     [-generateSteps]
                                     [-invPtrLoc <Path to oraInst.loc> ]
```



```

[-jre <LOC> ]
[-norestart ]
[-nonrolling ]
[-oh <ORACLE_HOME> ]
[ <Patch Location> ]

```

### Options

The following table describes the options available for the `rollback` command.

**Table B–4** `rollback` Command Options

Option	Description
-analyze	This option runs all the required prerequisite checks to confirm the patchability of the system without actually applying the patch or affecting the system in any way.
-database	The database option is used to specify the RDBMS home(s) to be patched. This option takes a comma-separated list of Oracle database names as input.
-generateSteps	Generate the manual steps that OPatchauto will perform during the rollback session.
-invPtrLoc	The <code>invPtrLoc</code> option is used to locate the Central Inventory Pointer File ( <code>oraInst.loc</code> ) . Input for this option is the path to the <code>oraInst.loc</code> file.
-jre	This <code>jre</code> option instructs OPatchauto to use the JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
-norestart	The <code>norestart</code> option instructs OPatchauto not to restart the Grid Infrastructure stack after patching.
-nonrolling	The <code>nonrolling</code> option instructs OPatchauto to run the patching session run in 'nonrolling' mode. Before the patching session can start, the following prerequisites must be met: <ul style="list-style-type: none"> <li>■ The stack on the local node must be running</li> <li>■ All remote nodes must be down.</li> </ul>
-oh	The <code>oh</code> option specifies the GI/RAC/DB home to work on. The GI/RAC/DB home specified takes precedence over the environment variable <code>ORACLE_HOME</code> .
<patch location>	The explicit path to the location of the patch. If the patch location is not specified, then the current directory is taken as the patch location.

**Example B–4 To roll back a patch on the GI Home and all RAC Homes**

```
<GI_HOME>/OPatch/patchauto rollback
```

**Example B–5 To roll back a patch on databases running from RAC Homes only**

```
<RAC_HOME>/OPatch/patchauto rollback -database db1,db2...dbn
```

**Example B–6 To roll back a patch from a software-only installation**

```

<RAC_HOME>/OPatch/patchauto rollback -oh <RAC_HOME>
OR
<GI_HOME>/OPatch/patchauto rollback -oh <GI_HOME>

```

## B.1.4 version

Print the version of the OPatch utility, dependent OPlan version, and the osysmodel version.

**Important:** OPatchauto must be run from the GI Home as a root user.

### Syntax

```
<GI_HOME>/OPatch/opatchauto version [-invPtrLoc <Path to oraInst.loc>]
                                     [-jre <LOC>]
                                     [-oh <ORACLE_HOME>]
                                     [-oui_loc <Custom OUI Location>]
                                     [-help] [-h]
```

### Options

The following table describes the options available for the `version` command.

**Table B-5** `version` *Command Options*

Option	Description
-invPtrLoc	The <code>invPtrLoc</code> option is used to locate the Central Inventory Pointer File ( <code>oraInst.loc</code> ) . Input for this option is the path to the <code>oraInst.loc</code> file.
-jre	This <code>jre</code> option instructs OPatchauto to use the JRE (java) from the specified location instead of the default location under GI/RAC/DB home.
-oh	The <code>oh</code> option specifies the GI/RAC/DB home to work on. This takes precedence over the environment variable <code>ORACLE_HOME</code> .
-oui_loc	The <code>oui_loc</code> option displays the custom OUI location from which OPatch runs.

---

---

# Glossary

## **Bundle Patch**

A bundle patch is a cumulative collection of fixes for a specific product or component. A patch of this type is released as needed depending on the product's requirements. You may also know a bundle patch as: maintenance pack, service pack, MLRs, cumulative patch, or update release.

## **Diagnostic Patch**

A diagnostic patch is designed to help diagnose or verify a fix or collection of bug fixes. You may also know a diagnostic patch as: test patch, Fix Verification Binary (FVB) or e-fix.

## **Interim Patch**

An interim patch provides a single bug fix, a collection of bug fixes, or a customer-specific security fix. They generally address specific bugs for a particular customer, and generally should not be applied unless instructed by Oracle Support to do so. You may also know an interim patch as: security one-off, exception release, x-fix, PSE, MLR, or hotfix.

## **MLR**

Merge Label Request. A bundle of patches fixing several bugs.

## **Patch Set**

The main way in which Oracle provides bug fixes in between releases. Oracle bundles a number of fixes, test them thoroughly together, and package them together for easy download and installation. They generally do not include new functionality and do not require a new certification. All of the fixes in the patch set have been tested and are certified to work with each other.

## **Patch Set Update**

A collection of proactive, stabilizing cumulative patches for a particular product version (base release or patch set). PSUs are cumulative and include all of the security fixes from SPU patches (formerly known as CPU), plus additional fixes.

## **Security Patch Update**

A security patch update is a cumulative collection of security-related bug fixes. Generally, security patch updates are released regularly. The security patch update was previously known as Critical Patch Update or CPU.

## **Singleton**

A patch with one bug fix.

**Patch**

A patch is a piece of code/software designed to fix problems with the existing code/software. This includes fixing security vulnerabilities and other bugs, and improving the usability or performance.

**Patch Conflict**

If a patch makes different changes to the same section of code that another OPatch modifies, then these two patches conflict, and only one of them can be installed (unless a merge or overlay patch is available).

**Superset Patch**

If a particular patch to be applied contains all of the fixes included in an already installed patch, plus additional fixes, then the patch with more fixes is a superset patch, and there is no conflict.

**Combination Conflict**

If a patch to be installed conflicts with more than one already installed patch, this is considered a combination conflict. In this case, OPatch will remove all conflicting patches then apply only the new patch.

**Critical Patch Update**

Critical Patch Updates (CPUs) are the primary means of releasing security fixes for Oracle products. CPUs are cumulative with respect to prior CPUs and generally contain only security fixes.

**Merge Patch**

A merge patch is one where multiple conflicting patches are combined into one integrated patch.

**Overlay Patch**

When an interim patch conflicts with a PSU, patch conflict resolution is achieved by providing a new patch that coexists with (and requires) the PSU patch. The new patch overlays the PSU, and the PSU is a pre-requisite for the overlay patch.

---

---

# Index

## A

---

Apply command  
    patch conflict behavior, 2-7  
automation, 3-1

## B

---

BPU, 1-5  
Bundle Patch Updates, 1-5

## C

---

Configuration Patching, 1-6  
CRS Error Codes, 5-2

## D

---

Datapatch, 5-6  
Diagnostic patches, 1-5

## E

---

Enterprise Manager, patching with, 1-6  
Error Messages, 4-1

## H

---

Homes, shared and non-shared, 3-2

## I

---

Interim patches, 1-5  
interim patches for security bug fixes, 1-5

## L

---

log file, debug, 5-8  
log file, locations, 5-8  
log file, naming conventions, 5-8  
log file, steps, 5-8  
Log File,Regular, 5-8  
log files, OPatchauto, 5-8  
Logging and Tracing, 4-5  
Logging and tracing, 4-5

## M

---

Manual Patching, 1-5  
Merge Label Request, 1-5  
MLR, 1-5  
Multiple Patches, applying, 2-3  
My Oracle Support, 1-2

## N

---

Napply command  
    patch conflict behavior, 2-7  
Non-rolling Mode, 3-2

## O

---

OPatch, 1-1  
    compare, A-4  
    conflicts, 2-5  
    lsinventory, A-5  
    lspatches, A-6  
    napply, A-7  
    nrollback, A-10  
    prereq, A-16  
    query, A-14  
    rollback, A-12  
    util, A-19  
    version, A-15  
OPatch Fails, 5-1  
OPatch Syntax, A-1  
OPatch Utilities, accessing, 1-8  
OPatchauto, 1-1  
    apply, B-1  
    resume, B-4  
    rollback, B-4  
    version, B-6  
OPatchauto Apply, 3-4  
OPatchauto Log Files, 5-8  
OPatchauto Patching Steps, 3-6  
OPatchauto Troubleshooting Architecture, 5-1  
OPatchauto, single node, 3-3  
OPERR, determining latest file, 4-2  
OPERR, latest, 4-1  
OPERR, running, 4-5  
Oracle Patches, types, 1-5  
Oracle Support Services, 1-4

ORACLE\_HOME, environment variable, 2-2

## **P**

---

Patch Application Modes, 3-2

Patch Application Prerequisites, 2-2

Patch Conflict, 2-4

Patch Conflict Detection and Resolution, 2-4

Patch Process, 1-2

Patch Set Updates, 1-5

Patch Types, not supported, 4-6

patches, Diagnostic, 1-5

patches, Interim, 1-5

Patches, obtaining, 1-2

Patchgen, 5-4

Patchgen Error Codes, 5-5

Patching Scenarios, 2-4

PSU, 1-5

## **R**

---

Rolling Mode, 3-2

Rootcrs.pl, 5-1

Rootcrs.pl Prepatch, 5-2

Running OERR, 5-2

## **S**

---

Security Patch Updates, 1-5

Single Patch, applying, 2-3

SPU, 1-5

System Patch

    patch, system, 1-5

System Reboot Request, 3-5

## **T**

---

Target Configurations, supported, 3-1

troubleshooting, 4-1