

Oracle Upgrade Companion

Version 2.20 3/27/2008

Welcome to the Oracle Upgrade Companion. This guide helps you upgrade your Oracle Database from Oracle9i Release 2 (9.2) to Oracle Database 10g Release 2 (10.2). The guide is not an automated tool but rather provides guidance for pre-upgrade, upgrade, and post-upgrade steps.

NOTE: This document is not a replacement for the Oracle Database 10g Upgrade Guide. It should be used as a companion document to the Oracle Database documentation.

This document is continually updated as new information becomes available. Please check back prior to your upgrade.

If advice or onsite assistance is needed during your upgrade, go to [Oracle's Upgrade Management Services](#) page for details. Oracle Upgrade Management Services is designed to simplify your unique upgrade challenges, no matter what stage you are at in the upgrade process.

The Upgrade Companion does not cover upgrading Oracle E-Business Suite or Siebel Business Applications. Please refer to your application documentation. For convenience, some common E-Business Suite and Siebel Business Application MetaLink notes are listed below. This is not a complete list.

Oracle E-Business Suite:

- [Note 216205.1](#): Database Initialization Parameters for Oracle Applications 11i
- [Note 396009.1](#): Database Initialization Parameters for Oracle Applications Release 12
- [Note 420649.1](#): DBUA from 10Gr2 Oracle Home comes with R12 RapidInstall is not working
- [Note 467778.1](#): Oracle 10gR2 (10.2.0.2) Database Preparation Guidelines for an E-Business Suite Release 12 Upgrade

Siebel Business Applications

- Coming Soon

Modifications

Version 2.20 March 27, 2008

Home Page

- Changed the order of the tabs.
- Added a link to Oracle's Management Services web page
- Clarified that the Upgrade Companion does not cover Application upgrades. Provided some common links to Application notes.

Best Practices Introduction Page

- Slightly modified the Usage section

Best Practices Upgrade Planning Page

- Added links to documents in the Technical Planning section
- Added a new documentation reference table for HA documents in the Technical Planning section
- Added details for the export/import and data copying upgrade methods in the Technical Planning section

Best Practices Prepare and Preserve Page

- Added additional links to documents

Best Practices Post Upgrade Page

- Modified the wording to stage 1 under Overview to make it clear that stage 1 needs to be performed until the upgrade of the TEST database is successful.
- Added new Post Upgrade Task section
- Added a new documentation reference table for HA documents in the Database Stability section

Behavior Changes Page

- Removed "Mutexes" from the Architecture section
- Removed "FILESYSTEMIO_OPTIONS" from the Initialization Parameters section. This parameter is not new to 10g
- Modified PARALLEL_ADAPTIVE_MULTI_USER under Initialization Parameters
- Modified Explain Plan Enhancements under Performance and Monitoring
- Moved Bind Peeking to the Optimizer section
- Added new link to the Note reference table under the Optimizer section
- Removed Creation of Index statistics during Create Index and Rebuild under the Optimizer section because it was the same as Statistics on Indexes
- Slightly modified the wording for Optimizer_secure_view_merging
- Added a new section for "Streams"

Recommended Patches Page

- Highlighted the last sentence in the first paragraph so that it would get the reader's attention
- Modified the mini patch information under the Microsoft Windows section
- Updated the list of patches for the UNIX platforms.

Version 2.10 March 3, 2008

Behavior Changes Page

- Modified Hash Group By aggregation enabled under the Architecture section.

Best Practices Prepare and Preserve Page

- Modified the Review Non-Default Initialization Parameters section under Preserve
- Modified the Preserving Performance Baselines and Statistics for Oracle9i Source Database section under Prepare

Best Practices Post Upgrade Page

- Page has been rewritten.

Version 2.00 February 8, 2008

Home

Modified the first paragraph to clarify the Oracle versions covered by this document.

Recommended Patches

Added more detail about the page in the introduction paragraph.

Behavior Changes Page

- Modified the description of the sections `Optimizer_mode`, `Optimizer_dynamic_sampling`, `Optimizer_secure_view_merging`, Automatic statistics gathering job, Changes in the default values for `DBMS_STATS` parameters, `DBMS_STATS` and parallel degree, System statistics, Bind Peeking, and Cost Based Transformations under Optimizer on the Behavior Changes page.
- Added the sections Statistics on Fixed Objects, Statistics on Indexes, and Restoring Statistics under Optimizer on the Behavior Changes page.
- Modified the description and moved the Hash Group By aggregation enabled section under Architecture on the Behavior Changes page.

Best Practices Upgrade Planning Page

Added note 369644.1 to the reference box on the right under the "Technical Planning" section.

Best Practices Post Upgrade Page

- Added MetaLink note 401436.1 to the Note reference box under DB Stability on the Best Practices Post Upgrade page.
- Corrected some links in the Note reference box on the Best Practices Post Upgrade page.
- Added hyperlinks to bugs listed in the Database Performance>Resolving Performance Regressions > If Regression is Due to a SQL statement...section.
- Under When All Else Fails...Going Back to the Previous Version section, modified link to the Oracle Database 10g Release 2 (10.2) Upgrade Guide to point to the Downgrading chapter.
- Added note 247611.1 and 463227.1 to the reference box on the right under the Database Performance section.
- Added Rule-Based Optimizer under the Optimizer section.

Contents

[Best Practices > Introduction](#)

[Introduction](#)

[Usage](#)

[Best Practices > Upgrade Planning](#)

[Documentation Review and Planning](#)

[Technical Planning](#)

[Quality Assurance](#)

[Known Issues](#)

[Best Practices > Prepare and Preserve](#)

[Prepare](#)

[Preserve](#)

[Best Practices > Upgrade](#)

[Pre-Upgrade Check List](#)

[Follow the Oracle Database 10g Release 2 Upgrade Guide](#)

[Best Practices > Post Upgrade](#)

[Overview](#)

[Post Upgrade Tasks](#)

[Database Stability](#)

[Database Performance](#)

[When All Else Fails...Going Back to the Previous Version](#)

[Obtaining Support](#)

[Behavior Changes](#)

[Architecture](#)

[Optimizer](#)

[Initialization Parameters](#)

[Performance and Monitoring](#)

[Administration](#)

[Streams](#)

[Patches Recommended](#)

[Operating System](#)

[Documentation](#)

[Documentation](#)

[Related Documentation](#)

[Database Features Documentation](#)

Feedback

We look forward to your feedback. Please email any comments, suggestion to help improve this guide, or any issues that you have encountered with the tool usage to Vickie.Carbonneau@oracle.com, Technical Advisor, Center of Expertise (CoE).

Best Practices > Introduction

Introduction

The Best Practices section is not a replacement for the Oracle Database Upgrade Guide but rather a companion document; emphasizing and elaborating on recommendations and requirements. The Best Practices were derived by Oracle technical staff and is an accumulation of real world knowledge and experience obtained while working with our customers.

Usage

The Best Practices tab is organized by the following major steps in the Upgrade Methodology:

- **Upgrade Planning:** Important information related to planning the DB configuration, and testing
- **Prepare and Preserve:** Information related to preserving and preparing the source environment for the Oracle Database 10g Upgrade
- **Upgrade:** Final reminders and information required for the actual upgrade
- **Post Upgrade:** Testing and analysis which should be performed after upgrading your test and production databases to Oracle Database 10g

The outlined steps should be validated and adjusted repeatedly in your test environment. The final upgrade plan and execution steps should run smoothly on test before proceeding to your production environment.

Here's a sample flow to illustrate the approach:

1. Upgrade Planning - Evaluate and document the plan for your test environment
 - The documented plan resulting for this step will be relevant for Test, Stage, and Production environments
2. Preserve and Prepare - Evaluate, document, and perform the steps to prepare your test environment
 - Decisions and steps outlined here will be relevant for both Test and Production environments
3. Upgrade - Upgrade your test environment
 - Document any lessons learned from this step to ensure

smooth execution when upgrading production

4. Post-upgrade - Use tips and techniques documented here to ensure your test environment is performing up to a standard required for production
5. At this point, test has been upgraded. Consider the following:
 - Have you adjusted your plan to include everything you learned from upgrading test?
 - This is important to avoid the same problems that were encountered in test when you upgrade production
 - Are you comfortable that you have a repeatable plan to upgrade production?
 - If not, upgrade test again
 - Are you comfortable that the system was tested adequately for functionality and stability and will adhere to all your performance and availability requirements?
 - This is critical since many problems encountered after upgrading to Oracle Database 10g can be avoided with proper testing
 - Have you tested your fallback plans and procedures?
6. Once you are comfortable that you can move on to upgrade stage or production execute steps 2 - 4 on that environment.

Best Practices > Upgrade Planning

Executing a well defined plan helps to mitigate risk and provides clear backup & recovery procedures in the event of a problem. The majority of the time spent during an upgrade project should be in the planning, preparation, and testing phases. The amount of detail defined in your plan and processes depends of the level of importance and time available to successfully complete the upgrade.

Documentation Review and Planning

The information provided below is intended as a supplement to the Oracle Upgrade Guide. The following list identifies items which should be considered when defining your upgrade plan.

- Review the detailed instructions provided in the [Oracle Database 10g Upgrade Guide](#).
- Carefully review the [Oracle Database Installation Guide for your specific Operating System](#); this guide describes how to install and configure Oracle Database 10g database.
- Review the New Features available in [Oracle Database 10g New Features Guide Release 2](#).
- Review the [Oracle Database Administrator's Guide 10g Release 2](#).
- Review the [Oracle Administrator Reference 10g Release 2 for UNIX Based Operating Systems](#) if you're upgrading in a UNIX/Linux environment.
- As each phase is being executed ensure the steps are well defined and carefully documented.
 - Take note of the expected timeframe for each phase.

Technical Planning

- [Determine the Upgrade Path](#)
- Choose an Upgrade Method - Oracle Database 10g Supports the following methods:
 - [Database Upgrade Assistant](#)
 - GUI Interface to Guide you through the process
 - Oracle's preferred method of upgrading
 - Advantages
 - Automates all tasks
 - Performs both Release and Patch set upgrades
 - Supports RAC, Single Instance and ASM
 - Informs user and fixes upgrade prerequisites
 - Automatically reports errors found in spool logs
 - Provides complete HTML report of the upgrade process
 - Command line interface allows ISVs to automate
 - Disadvantages
 - Offers less control over individual upgrade steps
 - [Manual Upgrade](#)
 - Command line upgrade using Oracle supplied SQL scripts and utilities ([Note:361889.1](#) provides a checklist for manual upgrades.)
 - Advantages
 - The DBA controls every step of the upgrade process
 - Disadvantages
 - More work
 - Manually check spool logs for errors
 - More error prone
 - More difficult to automate
 - [Export/Import](#)
 - Full or Partial Export followed by full or partial import into Oracle Database 10g
 - Advantages
 - Defragments the data
 - Restructures the database
 - Enables the copying of specified database objects or users
 - Serves as a backup archive
 - Disadvantages
 - Can take a long time
 - [Data Copying](#)
 - Copy data using SQL*Plus Copy Command
 - Advantages

Notes

- [Answers To FAQ For Restoring Or Duplicating Between Different Versions And Platforms](#)
- [Complete Checklist for Manual Upgrades to 10gR2](#)

HA Documentation

- [Oracle Database High Availability Best Practices](#)
- [Oracle Database High Availability Best Practices - Managing Outages](#)
- [Oracle Database High Availability Best Practices - Recovering from Unscheduled Outages](#)

- Defragments the data
- Restructures the database
- Enables the copying of specified database objects or users
- Serves as a backup archive
- Can copy only specified database objects or users
- Can copy data from one Oracle Database to another using database links
- Disadvantages
 - Can take a long time
- Ideally, the only change being made is the database upgrade.
- Document the steps to reproduce the environment and make sure to note any changes in the configuration.
- Determine if there are any known issues, Alerts, OS Patches, Database Patches, or Critical Patches Updates (CPUs) available. Ensure that all the critical patchsets & CPUs are applied.
 - Always apply all necessary patches in the target `$ORACLE_HOME` before doing the upgrade. This will omit an additional recompilation process and save time.
 - Check MetaLink [Note: 161818.1](#) - click on your target release (RELEASE column) in the colored table - click on the "Known issues and alerts" Note for your desired target patchset and check the ALERTS section as well as the UPGRADE ISSUES section for any known problems. Apply the recommended patches to your `$ORACLE_HOME`.
 - MetaLink certification: Always ensure that your desired OS-Oracle combination is certified. Go to MetaLink and check in the CERTIFY section.
 - Review "Behavior Changes" tabs in this document.
 - Review the "Patches Recommended" tab in this document for patches to apply after the upgrade.
 - Review MetaLink [Note: 461082.1](#) if any CPUs need to be applied.

Quality Assurance

Quality Assurance is a series of carefully designed tests to validate all stages of the upgrade process. Executed rigorously and completed successfully, these tests ensure that the process of upgrading the production database is well understood, predictable, and successful. Perform as much testing as possible before upgrading the production database. **Do not underestimate the importance of a test program.**

- Preserve configuration information, object / system statistics, and performance baselines (see *Prepare and Preserve step, Preserving Configurations and Statistics for Oracle9i Source Database*)
- The test database should be a complete reproduction of the production database using real data and not a small subset of production data. Make sure the environment is configured exactly the same as production. There are a number of ways to create a duplicate database you can recover using a backup, recreate the database or use RMAN to Duplicate database feature in Oracle9i and Oracle Database 10g, or Export/Import.
- Perform a test upgrade using a test database. The test upgrade should be conducted in an environment created for testing and should not interfere with the actual production database.
- Document, define, and test your back-out plan. Include checkpoints and success criteria at each phase. Determine clear ways to measure the success and failures and define what indicators will trigger a rollback. **Never skip this step!** Your data is important and fully testing your recovery scenarios is a priority. Do not just simulate your recovery plan you should perform all the necessary steps and document the results and timeframe needed to recover. This is discussed in more details under the Prepare and Preserve step, Preserving the Database.
- Ensure you budget enough time to test all applications. This not only includes your primary application but also secondary applications, such as PL/SQL code, Shell Script, any APIs, Pro*C, all Interfaces, and any third party administrative tools.
- Perform load testing to determine if the database can handle the load; the database performance must be equal to or exceed the performance of the peak production workload. Record the steps to reproduce or automate the load testing; log and record your results. For load testing best practices see [Note: 466452.1](#).
- Review your plan during the testing process to ensure all the steps are being documented and make sure variations and changes are documented. Failing to document a change in the plan may lead to an undesirable result. Capture all changes in a "Change Log" and the entire upgrade process should be evaluated and improved by the change control committee.
- The testing process should be executed as many times as needed until the final result is a successful upgrade. Once you have successfully upgraded the database and tested the test database do another upgrade.

How-To

- [RMAN 'Duplicate Database' Feature in Oracle9i/Oracle 10G](#)
- [Creating a Duplicate Database on a New Host](#)
- [How To Create A Production \(Full or Partial\) Duplicate On The Same Host](#)

Known Issues

Below are known issues which could be encountered during the upgrade process. These issues may prevent a successful upgrade. For behavior changes and recommended patches please click on the *Behavior Changes* or *Patches Recommended* tabs.

- If your database was created as a 32-bit database (even though it might currently be 64-bit) and is now being upgraded to 10.2.0.3 64-bit, you could receive the following errors `ORA-600 [22635]` and `ORA-600 [KOKEIIX1]` while upgrading or patching databases to 10.2.0.3 [Alert: 412271.1](#)

Best Practices > Prepare and Preserve

Defining, implementing and managing a contingency plan is extremely important step during the upgrade process. Mission critical enterprises require a return to normal operations more quickly today than ever before. Accordingly, system availability is dependent on how well you prepare for outages. Planning and practicing for the unexpected issues helps to ensure the upgrade to the new Oracle Database 10g will be successful.

Prepare

Preparing the database before the upgrade begins will help reduce any unforeseen errors or circumstances that prevent the upgrade from completing, such as out of space errors. There are a number of things to consider before the upgrade begins, including fallback planning, compatible parameter setting considerations and database configuration settings.

Execute the Pre-Upgrade Information Tool

Preparing the current database for a successful upgrade entails running the Pre-Upgrade Information Tool, detailed in chapter 3 of the Upgrade guide. This SQL script checks the following:

1. Database configuration: Determines if the logfiles and datafiles are sized adequately for the upgrade
2. Initialization parameters: Reports which init.ora parameters need changing, replacing or removing before the upgrade
3. Components: Which installed components will be upgraded
4. Miscellaneous Warnings: Any other situations requiring attention before the upgrade
5. Required tablespace: Ensure that the SYSAUX tablespace is created in the current database BEFORE the upgrade is carried out.

Implement the recommendations reported by this script before performing the upgrade.

COMPATIBLE Initialization Parameter

The COMPATIBLE initialization parameter needs a special mention here because it has consequences if the database needs to be downgraded. Once the database has been upgraded and the compatible parameter has been set to 10.1 or 10.2, once restarted the datafiles, controlfiles and online logfiles are updated to the new version. This in turn will prevent the database from being downgraded in the future. Any attempt to downgrade the database will report an error:

```
SQL> STARTUP DOWNGRADE;  
ORACLE instance started.
```

```
Total System Global Area 436207616 bytes  
Fixed Size 2029528 bytes  
Variable Size 327157800 bytes  
Database Buffers 104857600 bytes  
Redo Buffers 2162688 bytes  
ORA-00201: control file version 10.2.0.0.0  
incompatible with ORACLE version 9.2.0.6.0  
ORA-00202: control file: '/u01/oradata/B920/  
control01.ctl'
```

When this occurs the only way to downgrade the database is to restore the database from the backup taken before the database was upgraded or to use any alternate strategies in place like Streams, Data Guard or Export/Import. For further details on planning a fallback strategy, read the 'When to Fallback' section below.

Because of the inability to downgrade the database once it has been opened with the new compatible parameter, it is recommended to leave the parameter set to the 9.2 value until the newly upgraded database performance and functionality is acceptable. At that time, the compatible parameter can be reset to the new, higher version and any new features that require COMPATIBLE to be 10.1 or higher can begin to be used.

More details on the compatible setting can be found in chapter 5 of the [Upgrade Guide](#).

Review Non-Default Initialization Parameters

It is common to change initialization parameters away from their default values to adapt an instance to a particular workload or sometimes to put a workaround in effect (via an EVENT or underscore parameter). When upgrading a database it is important to review these parameters (especially EVENTS) and determine if they are no longer needed or can cause adverse effects in the new version - reduce non-default parameters to the bare minimum possible when upgrading.

Check which parameters have non-default values on your Oracle9i database by executing this query:

```
col name format a30
col value format a60
set linesize 130
set pagesize 2000

SELECT KSPPINM "Name", KSPFTCTXVL "Value"
FROM X$KSPPI A, X$KSPPCV2 B
WHERE A.INDX + 1 = KSPFTCTXPN
AND KSPFTCTXDF <> 'TRUE'
ORDER BY 2;
```

Contact Oracle Support Services if you find an undocumented parameter (begins with "_") or an EVENT set that you are unsure about.

When to Fallback

Prior to the upgrade, the customer should have a fallback strategy in case Service Level Agreements(SLAs) are not met and resolving the issues on the upgraded system cannot be done within some agreed upon time. The customer should answer the following questions to understand when fallback should be considered.

For the most part, only severe performance regressions that cannot be resolved in a timely fashion will require fallback.

- What severe events constitute the need for a fallback?
- What's the maximum time before fallback is initiated?
- What's the target Recovery Time Objective(RTO) and Recovery Point Objective(RPO) to complete the fallback?
- What fallback options have been tested and do they meet the above SLAs?

For information on the fallback plan and fallback options, see the *Preserving the Database - Fallback Plan* under the *Preserve* section below.

Recommendations to Avoid Common Pitfalls

1. **Set the appropriate kernel parameters for your system as defined in your operating-system specific [Oracle Database 10g Installation Guide](#).**
2. **Verify that all OS patches and packages are installed as defined in your operating-system specific [Oracle Database 10g Installation Guide](#).**
3. **Additional space is needed during the upgrade to avoid unable to extend errors, set `AUTO EXTEND ON MAXSIZE UNLIMITED` for the `SYSTEM` and `SYSAUX` tablespaces.**
4. **You may require larger shared memory pool sizes in some cases. See [Oracle Database Reference](#) for information about shared memory initialization parameters.**
5. **If you are using Database Upgrade Assistance (DBUA) as the method to upgrade, make sure that there is sufficient OS temp space.**
6. **If you are use dblinks, you should document your dblink creation scripts as a precaution in case you need to downgrade. During the upgrade to Oracle Database 10g, the upgrade process will encrypt connection passwords. To revert back you will need to drop and recreate the dblinks.**
7. **Check and ensure there are no invalid objects in `SYS` and `SYSTEM` user schema.**

If there are invalid objects you can recompile them by logging in as a `SYSDBA` user and running the `utlrbp.sql` located in the `$ORACLE_HOME/rdbms/admin` directory. This script will attempt to recompile the invalid object and any dependencies. This script can be run multiple times until all objects have been compile. Additionally the Pre-Upgrade Information Tool located `$ORACLE_HOME/rdbms/admin/utlu102i.sql` checks for invalid objects.

8. **Disable all `DBMS_JOBS`, Batch, AT, and Cron Jobs before starting the upgrade.**

Please note that using `STARTUP UPGRADE` will automatically disable database jobs; however, any OS level or third party tools that spawn jobs at the OS level that then connect to the DB or do cleanup tasks must be manually disabled.

9. In Oracle 10.2 the `CONNECT` role only includes `CREATE SESSION` privilege.

If you have user or roles that require privileges other than `CREATE SESSION` then document the user and roles and grant the specific privileges after the upgrade. A `WARNING` is provided if you run the Pre-Upgrade Information Tool located `$ORACLE_HOME/rdbms/admin/utlu102i.sql`.

10. Run the Pre-Upgrade Information Tool located `$ORACLE_HOME/rdbms/admin/utlu102i.sql` which validates the source database is ready to be upgraded. Any area that does not meet the requirements must be corrected.

11. Make a backup copy of the following files; `init.ora`, `tnsnames.ora`, etc...

12. Upgrade or recompile Pro*C programs.

One common step that is over looked is forgetting to upgrade or recompile Pro*C programs. If you are using OCI make sure to review [Oracle Call Interface Programmer's Guide - Introduction and Upgrading](#) which contains information on upgrading to a new release of OCI.

Preserve

It is important to preserve the current database before the upgrade begins so that a known good state of the database can be restored in the event of a failure during the upgrade process. Having a good fallback plan in place can prevent significant down time and data loss in the event of a failure. Recording current performance data is critical to diagnosing any performance issues that arise once the upgrade is completed. Without pre-upgrade data it will not be possible to determine why performance characteristics have changed.

Preserving Performance Baselines and Statistics for Oracle9i Source Database

Documentation

- [Backup and Recovery Basics Guide](#)

How-To

- [Managing CBO Stats during an upgrade from 9i to 10g](#)
- [Recording Explain Plans on 9i before an upgrade to 10g](#)
- [Recommended Method for Obtaining 10046 trace for Tuning](#)

Scripts and Tools

Proper performance testing is key to a successful upgrade. This section discusses what needs to be done to properly capture performance data at all stages of the upgrade process.

1. Capturing Performance Baselines Before Upgrading

It is very important to capture performance baselines before and after the upgrade process (in both TEST and PRODUCTION). These baselines will allow us to detect a performance regression on the TEST system and perhaps later in production. These baselines will be captured in three ways:

- 1. Unit tests: specific queries, transactions, and jobs that are important to the business***
- 2. Load tests: a load simulation that runs important business activities at similar user levels and concurrency rates as are run on the production system***
- 3. Production workloads: Actual production workloads captured prior to the upgrade from Oracle9i and after completing the upgrade to Oracle Database 10g***

Baselines, Step-by-step

- 1. Choose the most important transactions, batch jobs, or queries (we'll call these "activities") that must NOT be negatively impacted by the upgrade.***
- 2. Determine acceptable response times, throughput, and/or job execution times that cannot be exceeded***
- 3. Construct unit tests for the activities.***
- 4. Construct accurate load tests by choosing the mix of activities, number of users, and activity rate that simulates production workloads.***

On the TEST system, perform the following BEFORE upgrading:

- 5. Capture performance baselines for the unit tests. This includes:***
 - o Execution timing for each activity. We will use this later to see if a job is taking longer after the upgrade.***
 - o Statspack snapshot just before and just after the critical job or activity. Take snapshots at level 7 to capture execution plans and segment statistics. The thresholds for capturing SQL statements must be set low to ensure capturing all SQL related to this activity. Additional snapshots may be taken during the activity at 1/2 hour intervals.***

- [OS Watcher](#)
- [OSWg](#)
- [LTOM](#)

White Papers

- [Maximum Availability Architecture \(MAA\)](#)
- [Determining CPU Resource Usage for Linux and Unix](#)
- [Measuring Memory Resource Usage for Linux and Unix](#)

- [Extended SQL tracing](#) (event 10046 level 8); ideally starting the trace just before the activity and ending it just after
6. Run the load tests and capture the following data:
- Operating system metrics ([OS Watcher](#) is recommended)
 - [Statspack snapshots](#) taken just before and just after the workload (at level 7, with low SQL capture thresholds). Additional snapshots may be taken at 1/2 hour intervals.
 - For easier analysis later, we recommend installing [LTOM](#) to capture system profiles during the load tests. Use [LTOMg](#) to graph the data collected by LTOM.
 - Capture application level response times during the test (some load testing tools will capture this for you).

On the PRODUCTION system, perform the following BEFORE upgrading:

7. Capture the following metrics when critical jobs or peak loads are running:
- Operating system metrics ([OS Watcher](#) is recommended)
 - [Statspack snapshots](#) taken just before and just after the workload (at level 7, with low SQL capture thresholds). Additional snapshots may be taken at 1/2 hour intervals.
 - For easier analysis later, we recommend installing [LTOM](#) to capture system profiles. Use [LTOMg](#) to graph the data collected by LTOM.

NOTE: If the Oracle9i Release 2 production system is already resource-constrained (CPU, memory, and/or I/O) as seen in the OSW or OSWg output, then the Oracle Database 10g system will also be resource-constrained - consider a capacity increase before upgrading. For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: [Determining CPU Resource Usage for Linux and Unix](#) and [Measuring Memory Resource Usage for Linux and Unix](#).

This data will be used for performance comparisons after the

upgrade (See *Best Practices > Post Upgrade > Database Performance*).

2. Additional Data to Collect

1. **Run Remote Diagnostic Agent (RDA) [Note:314422.1](#) on your production database and save for future reference.**
2. **See [Managing CBO Stats during an upgrade from 9i to 10g](#) before upgrading from Oracle9i to Oracle Database 10g Release 2 to see what should be done with object statistics.**
3. **Backup Performance-related data**
 - **Export the PERFSTAT user to preserve statspack data.**
 - **Export the OUTLN user**
 - **Export the statspack user**
 - **Backup collected performance data**

Preserving the Database - Fallback Plan

Taking a backup of the current database before an upgrade is carried out provides the ability to restore the pre-upgraded database if problems arise during the upgrade process that prevent the current database from being opened.

The ideal situation is to upgrade a copy of the database leaving the current database in place so that should any serious problems arise it is possible to revert back to the pre-upgraded database with little down time. This method would obviously require twice the amount of disk space than the current database occupies.

If the database is too big to copy, the upgrade will be done in place necessitating a good backup before the upgrade begins. If no backup is taken and the upgrade process fails, the amount of downtime to restore an older pre-upgrade database and recover must be taken into consideration.

The database can be backed up using a cold backup (the database is shutdown) or a hot backup (database remains open) and remember to include the init.ora. In the event of having to carry out a recovery from a failed upgrade attempt, if the database is running in ARCHIVELOG mode the pre-upgraded database can be recovered up until the database was started with the UPGRADE option. It also doesn't matter if the database is backed up using incremental or full RMAN backups, so long as the restoration and recovery times are acceptable from previous testing.

As a note of caution, the backup strategy used before the upgrade

should be one that has been tested and confirmed to recover the database in case of failure. Now is not the time to introduce a new backup strategy that has never been tested.

Further information on taking full database backups can be found in the [Backup and Recovery Basics guide](#).

As well as taking a backup of the database, it is recommended to take a backup of the oraInventory directory so that it can be restored in the event of a corruption occurring during the installation of the new Oracle software release. This may prevent further attempts at installing the new software or other patch installations of other releases installed on the machine. The backup can be done by simply copying the oraInventory directory (with or without compression). Taking a backup of the current ORACLE_HOME directory will also provide a stable installation that can be reinstated should the need to fallback to the pre-upgraded version be required.

1. Fallback Options

Prior to changing database compatibility

Prior to changing database compatibility, you have the following fallback options. These procedures need to be tested thoroughly. Note that "data loss" is only possible for any of these options if the application has started post-upgrade and began making changes.

OPTIONS	STEPS	CONSIDERATIONS
Downgrade	<ol style="list-style-type: none"> 1. Shutdown database 2. Downgrade 3. Restart 	<ul style="list-style-type: none"> • Zero Data Loss • Database integrity must be in place • Refer to upgrade guide
Streams	<ol style="list-style-type: none"> 1. Switch back to your replica 	<ul style="list-style-type: none"> • Zero Data Loss • Streams requirement • Refer to Streams guide • < 1 minute RTO
Restore to backup	Restore, Recover and Activate	<ul style="list-style-type: none"> • Data Loss • Refer to RMAN guide

After changing database compatibility

After changing database compatibility, you have fewer options.

OPTIONS	STEPS	CONSIDERATIONS
Streams	<ol style="list-style-type: none"> 1. Switch back to your replica 	<ul style="list-style-type: none"> • Zero Data Loss • Streams requirement • Refer to Streams guide • < 1 minute RTO
Export/Import	<ol style="list-style-type: none"> 1. Export or Unload all changes 2. Import changes 	<ul style="list-style-type: none"> • Zero Data Loss • Very time consuming • Existence of Read Only tablespaces can reduce work data by skipping those tables • Refer to utilities guide
Restore to backup	Restore, Recover and Activate	<ul style="list-style-type: none"> • Data Loss • Refer to RMAN guide

Best Practices > Upgrade

You are now ready to upgrade your environment to Oracle Database 10g. Before doing so, this section provides a list of all the key tasks and questions which should have been previously addressed. While reviewing this list, any items which have not been addressed should be revisited before proceeding to the actual upgrade step. Once you are satisfied that all items on the list have been addressed, it's time to begin your upgrade.

Pre-Upgrade Check List

At this point you should have:

- Reviewed the Oracle Database 10g Release 2 Upgrade Guide?
- Read the Oracle Database Readme for Oracle Database 10g Release 2?
- Read the Oracle Database Installation Guide for your specific Operating System?
- Read the Oracle Administrator Reference 10g Release 2 for UNIX Based Operating System?
- Make sure you have selected the best Upgrade Method for your business needs and make sure your planning and testing supports your decision.
- A well defined test plan that entails an Upgrade Test Plan, Functional Test Plan, Integration Test Plan, Performance Regression Test Plan, and Backup Strategy with Backup Test Plan.
- Executed the Pre-Upgrade Information Tool (`$ORACLE_HOME/rdbms/admin/utlu102i.sql`) and resolved any reported errors?
 - When this script runs it checks and displays a warning regarding:
 - displays the global database info and checks the compatibility initialization parameter
 - checks for redo logfile sizes less than 4MB
 - all tablespaces are checked for adequate space prior to upgrade
 - displays a list of parameters which need to be corrected prior to upgrade
 - displays a list of deprecated and obsolete parameters in Oracle Database 10g
 - provides a list of component features that will be upgraded
 - displays a list of all invalid objects and other miscellaneous warnings
- Verified that the desired OS-Oracle combination is certified by reviewing the Certify section in metalink.?

- Verified the kernel parameters are set according to the Oracle Database 10g Installation Guide?
- Upgraded your Operating System and/or installed the necessary Operating System Patches per the Oracle Database 10g Installation Guide?
- Collected object and system (if applicable) statistics and performance baselines from the Oracle9i production system as described in the *Prepare and Preserve step, Preserving Configurations and Statistics for Oracle9i Source Database*.
- Created a duplicate production environment for Testing Purposes?
- Applied all the necessary Critical Patches?
- Taken a backup of your Test Environment to test your Contingency Plan?
- Documented and tested all fallback and repair scenarios?
- Scheduled the downtime required for backing up and upgrading the production database?

Follow the Oracle Database 10g Release 2 Upgrade Guide

Now that you have successfully completed all the pre-upgrade steps including defining, developing, and documenting your test plans, it is time to test the process. During the testing process keep in mind that you will also need to reproduce these steps during the production upgrade. Continue testing the upgrade process until you can successfully upgrade without any errors; not until then should you attempt to upgrade your production environment. Follow the steps defined in the [Oracle Database Upgrade Guide 10g Release 2 \(10.2\)](#)

Best Practices > Post Upgrade

Overview

This section discusses important tasks to ensure database stability and performance. Recall (from *Best Practices > Prepare and Preserve > Performance Baselines...*) that the upgrade process will occur in two stages:

1. Repeatedly on a TEST system (ideally configured exactly as the production system) until successful
 - Choose queries, transactions, and jobs that are critical to the business and must be tested. Determine response time and/or throughput targets for each activity to be tested.
 - Construct unit tests to test the performance of each activity
 - Construct load tests to test the performance of each activity under load conditions that simulate production peaks
 - Run the unit and load tests before upgrading to collect performance and application baselines
 - Upgrade the TEST database (applying necessary patches, etc)
 - Run the unit tests and load tests again after the upgrade to the TEST database
 - Compare the post-upgrade results to the pre-upgrade baselines
 - Resolve any performance regressions in TEST before attempting to upgrade production
2. Then, on PRODUCTION (if the upgrade on TEST was successful)
 - Collect performance baselines during critical and peak loads BEFORE upgrading the PRODUCTION database
 - Upgrade the production database
 - Collect performance and application metrics from the production database AFTER the upgrade
 - Compare the production database performance using data collected before and after the upgrade
 - Investigate and resolve any regressions

Post Upgrade Tasks

This section discusses important tasks to ensure database stability and performance. Recall (from *Best Practices > Prepare and Preserve > Performance Baselines...*) that the upgrade process will occur in two stages: More tasks will need to be performance after upgrading a database. Review each of the following:

1. [Oracle Database Upgrade Guide 10g Release 2 - Chapter 4: After Upgrading a Database](#)
2. Recommended Patches tab in this document
3. MetaLink [Note: 461082.1](#) - Do I Need To Run catcpu.sql After Upgrading A Database?

Database Stability

Ensure the database is stable: no crashes, ORA-7445, ORA-600 errors, or unexpected trace files. This should be verified using your test suite that runs business-critical jobs and queries and can be done along with the performance regression tests discussed below in the *Database Performance* section. Re-check for signs of instability during and after load testing for performance (see below).

These stability checks should also be done immediately after the production upgrade goes "live" (but problems should've been already caught during testing).

All Databases

1. Review the *alert.log* since startup and see if any ORA-7445, ORA-600, or other errors are present.
2. Check your application logs and look for unexpected errors.

RAC Only

1. Ensure nodes are stable and no evictions occur.
2. Review the CRS, CSS, and EVM logs; grep for strings like "FATAL" or "ERROR"
3. If any errors or problems are found, please run *diagcollection.pl* as discussed in the [CRS 10g R2 Diagnostic Collection Guide](#)

Instability issues are usually caused by bugs or an incorrect configuration (using uncertified components). Ensure you have installed version 10.2.0.3 and applied the recommended patches (see the Recommended Patches tab) properly. If all patches have been applied, see the [list of patches](#) that are scheduled to be included in the 10.2.0.4 patchset. Search in the sections *Instance May Crash* , *Process May Dump (ORA-7445) / Abend / Abort*, *Internal Error May Occur (ORA-600)*.

Double-check [Metalink's Certify](#) to be sure you have installed certified

Notes

- [Oracle Clusterware consolidated logging in 10gR2](#)
- [10.2.0.4 Patch Set - List of Bug Fixes by Problem Type](#)

Scripts and Tools

- [CRS 10gR2 Diagnostic Collection Guide](#)
- [Troubleshooting CRS Reboots](#)

HA Documentation

- [Oracle Database High Availability Best Practices](#)
- [Oracle Database High Availability Best Practices - Managing Outages](#)
- [Oracle Database High Availability Best Practices - Recovering from Unscheduled Outages](#)

components.

Database Performance

Perform Post-Upgrade Cost-Based Optimizer (CBO) Management Tasks

It is **critical** to properly manage CBO statistics after an upgrade to Oracle Database 10g. There are many changes in Oracle Database 10g with regard to the CBO and having proper statistics gathered is essential to getting good performance.

See [Managing CBO Stats during an upgrade from 9i to 10g](#) after upgrading from Oracle9i to Oracle Database 10g to determine what should be done with statistics.

After you have addressed the CBO statistics, you are ready to begin validating the performance of the test system, or in the case of production, to begin monitoring production to catch any regressions that might have occurred.

Checking Database Performance

It is very important to check the performance of the database after upgrading the test and production databases. In TEST, this is accomplished by repeating the unit tests and load tests that were defined and executed before the upgrade (and discussed in the Best Practices > Preserve section). The unit tests should be done first so that any regressions can be addressed before going further.

After the unit tests are successful, the load tests should be performed and compared to the results of the load tests performed before the upgrade. Any regressions should be addressed before going further.

The production database should not be upgraded until performance regressions found in TEST are understood and resolved.

See the following sections for additional details.

Case Studies

- [Resolving High CPU Usage on Oracle Servers](#)

Documentation

- [Oracle Database 10g Release 2 \(10.2\) Upgrade Guide > Ch. 4: After Upgrading](#)
- [Oracle Performance Tuning Guide, SQL Tuning Overview](#)
- [SQL Tuning Advisor](#)

How-To

- [Recommended Method for Obtaining 10046 trace for Tuning](#)
- [How to Identify Resource Intensive SQL for Tuning](#)
- [Managing CBO Stats during an upgrade from 9i to 10g](#)
- [Recording Explain Plans on 9i before an upgrade to 10g](#)
- [Oracle Performance Diagnostic Guide, Query Tuning](#)

Notes

- [Knowledge Browser, see Upgrade and Migrations](#)
- [Knowledge Browser, see Query Tuning](#)
- [Known RMAN Performance Problems](#)
- [Oracle10g RMAN Recovery Catalog Known Performance Issues](#)

1. Checking the Performance of the TEST System with Unit Testing

The unit tests will check the performance of specific queries, transactions, and jobs that are important to the business (decided in Best Practices > Preserve > Preserving Performance Baselines... section) . The results of these tests will be compared to what was obtained before the upgrade in TEST. In summary, the test should capture:

1. **Execution timing for each activity.**
 - **Compare the elapsed times of each query, transaction, or job.**
 - **Investigate any activities which have increased elapsed times or lower throughput (beyond what the business can tolerate).**

2. **[Statspack snapshots](#)(or AWR snapshots, included in the Diagnostic Pack) just before and just after the critical job or activity.**
 - **Take snapshots at level 7 to capture execution plans and segment statistics. The thresholds for capturing SQL statements must be set low to ensure capturing all SQL related to this activity.**
 - **Additional snapshots may be taken during the activity at 1/2 hour intervals.**
 - **Generate an AWR report (or statspack if you are not licensed to use AWR) for periods/activities that are of interest**
 - **Add the total time of all "Timed Events" in the reports and compare to similar data gathered before the upgrade. If the upgraded database shows higher total time, investigate which timed event (CPU or wait) is higher and determine the cause.**
 - **Compare other resource usage such as total logical and physical reads, and total redo generated (see the Load Profile section). Investigate the reason for higher resource consumption (keep in mind that Oracle Database 10g may use more resources depending on which new features are being utilized)**
 - **Compare Top SQL statements (see the "Top SQL by..." sections). Look for large changes in elapsed time, CPU time, buffer gets, and physical reads. Investigate any statements which have regressed.**

3. **[Extended SQL tracing](#) (event 10046 level 8)**

Scripts and Tools

- [OS Watcher](#)
- [OSWg](#)
- [LTOM](#)

White Papers

- [Best Practices for Load Testing System Upgrades](#)
- [Determining CPU Resource Usage for Linux and Unix](#)
- [Measuring Memory Resource Usage for Linux and Unix](#)

- **start the trace just before certain activities and end it just after (for details on how to obtain these trace files, see [Recommended Method for Obtaining 10046 trace for Tuning](#) and for Oracle Database 10g see [DBMS_MONITOR package](#)).**
- **Obtain a [TKProf report](#) of the test before and after the upgrade**
- **Scroll down to the summary sections at the bottom and compare the total times, CPU times, "disk" (physical reads), "query" (consistent gets), and "current" (current gets).**
- **If there is a difference that needs to be investigated, rerun TKProf and sort the report by whichever metric significantly changed.**

If performance has regressed, see the section below called, "Resolving Performance Regressions". If performance is acceptable, continue to the load testing section below.

2. Checking the Performance of the TEST System with Load Testing

This validation looks at how performance has changed when the system is under load. Load testing is a non-trivial exercise that must be approached with care. Be sure that you account for the following:

- 1. Pre- and post-upgraded systems must be identical (same processor architecture etc), otherwise its difficult to compare and attribute changes to the upgrade itself.**
- 2. Pre- and post-upgraded systems must have identical and predictable workloads. A repeatable script or utility that produces a realistic workload on the database is essential. The system must not have non-Oracle tasks or transient workloads that may interfere with the system, skewing the performance test results and making comparisons worthless. However, if the system does have many transient jobs, at some point they should be enabled and the entire system load tested and compared to a similar workload before and after upgrading the TEST system.**

Please see the white paper, [Best Practices for Load Testing System Upgrades](#), for more details.

Data Collection

It is assumed that you have collected the following data for

both the pre-upgrade and post-upgrade TEST system during load testing (see also Best Practices > Preserve > Performance Baselines...):

- **[LTOM Profiles](#) (captures OS and database activity over time)**
- **OS data (sar, vmstat, or [OS Watcher](#))**
- **Database data ([Statspack](#) , AWR) for system-wide evaluation (1/2 hour snapshots, level 7 if statspack), and [extended SQL trace](#) (10046) / TKprof for individual batch jobs or key sessions/activities.**

Pre/Post Upgrade Performance Comparison

After performing the load tests, compare the post-upgrade results to the pre-upgrade results. Consider the following:

- **For batch jobs:**
 - **Note the total execution time for all jobs and compare to the total for the previous version. If the difference is more than your business can tolerate, determine which jobs are the reason for the increased time.**
- **For online transactions, note the average response time for key operations or pages (e.g., order entry) and compare to the previous version.**
- **Compare OS data during periods of similar activity and load for pre-upgrade and post-upgrade**
 - **This is most easily accomplished by using [LTOM](#) profiles with LTOMg's comparison feature**
 - **If you collected pre- and post-upgrade OS data using [OS Watcher](#), you can now graph the data using [OSWg](#) and compare the differences.**
 - **For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: [Determining CPU Resource Usage for Linux and Unix](#) and [Measuring Memory Resource Usage for Linux and Unix](#).**
- **Compare "database time" (total accumulated service and wait time for foreground sessions in the database) during periods of similar activity and load**
 - **Using the Oracle9i Release 2 statspack data, use statspack reports during critical activity times and estimate DB time by adding the top 5 or 10 wait times plus the CPU time..**
 - **In Oracle Database 10g, collect an AWR report of**

similar time frame to the Oracle9i Release 2 database (otherwise you'll need to scale the resulting values), and examine the value of DB Time in the Time Model section of the report.

- **If the DB Time is similar between Oracle9i Release 2 and Oracle Database 10g, then the upgrade has not regressed the application's performance. Otherwise, you will need to identify what has regressed and why.**

If a performance regression is found (jobs or transactions take longer or system uses more CPU, etc), see the next section below called, "Resolving Performance Regressions".

Otherwise, if performance is acceptable, you have finished performance testing on the TEST system.

3. Checking the Performance of the PRODUCTION System After an Upgrade

It is assumed that you have collected the baseline performance data for both the pre-upgrade and post-upgrade PRODUCTION systems during peak/critical workloads (see also Best Practices > Preserve > Performance Baselines...). Now, ensure that you are collecting performance data on the production system after upgrading it (assuming the performance of the test system has been checked and any regressions resolved). To summarize, the following should be collected:

- **[LTOM](#) Profiles (captures OS and database activity over time)**
- **OS data (sar, vmstat, or [OS Watcher](#))**
- **Database data ([Statspack](#), AWR) for system-wide evaluation**

Pre/Post Upgrade Performance Comparison

Compare the performance of the production database before and after the upgrade using the LTOM, OSWatcher, and statspack/AWR data collected.

- **Compare OS data during periods of similar activity and load for pre-upgrade and post-upgrade**
 - **This is most easily accomplished by using LTOM profiles with LTOMg's comparison feature. Look for places where the comparison graphs diverge (in an undesired way).**
 - **If you collected pre- and post-upgrade OS data using [OS Watcher](#), you can now graph the data using [OSWg](#) and compare the differences.**

- *For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: [Determining CPU Resource Usage for Linux and Unix](#) and [Measuring Memory Resource Usage for Linux and Unix](#).*
- *Compare "database time" (total accumulated service and wait time for foreground sessions in the database) during periods of similar activity and load*
 - *Using the Oracle9i Release 2 PRODCUTION statspack data, use statspack reports during critical activity times and estimate DB time by adding the top 5 or 10 wait times plus the CPU time..*
 - *In Oracle Database 10g Release 2 PRODUCTION, collect an AWR (or statspack) report of similar time frame to the Oracle9i Release 2 database (otherwise you'll need to scale the resulting values), and examine the value of DB Time in the Time Model section of the report.*
 - *If the DB Time is similar between Oracle9i Release 2 and Oracle Database 10g Release 2, then the upgrade has not regressed the application's performance. Otherwise, you will need to identify what has regressed and why.*

NOTE: *Its very important to be sure you are comparing similar workloads, otherwise the comparison may give false positive or negative results.*

If performance has regressed, see the section below called, "Resolving Performance Regressions".

4. Capture Performance Metrics During Normal Operation

After the performance of the database has been verified as acceptable post-upgrade, one should still routinely gather performance metrics to enable future baseline comparisons and for performance troubleshooting.

- 1. If you are not licensed to use AWR, then install [Statspack](#) and set the default level to 7*
- 2. Schedule Statspack or AWR snapshots every 30 minutes*
- 3. Gather OS statistics (consider using [OS Watcher](#) at 1 minute intervals).*
- 4. Oracle Support recommends installing the [LTOM](#) utility proactively to help troubleshoot problems in the future*

Resolving Performance Regressions

The suggestions in this section may be used when resolving performance problems after upgrading test or production.

1. Overall Approach

If you are licensed for the [Diagnostics and Tuning Packs](#), use the [Automatic Database Diagnostic Manager \(ADDM\)](#) to diagnose performance problems. Otherwise:

1. **Determine problem scope - is the performance system-wide or is it localized to certain users/applications**
 - If the scope is system-wide, its best to start with a statspack or AWR report and drill-down into the problem by analyzing the timed events.
 - If the problem is found with a particular session, query, or transaction, then a 10046 trace of a session having the problem will be useful. Of course, if you are licensed for the EM Tuning pack, use the [SQL Tuning Advisor](#) to investigate the SQL.
2. **Determine where time is being spent - i.e. CPU or wait events. In either case, the typical drill down will be SQL.**
 - If CPU, find the cause of the increase in CPU, typically SQL statements consuming CPU.
 - If wait events, is it an increase in I/O (includes block requests from remote caches) or is it a serialization point (latches, etc.). Look for SQL statements with long elapsed times
3. **Determine if the increase is due to**
 - Increased execution frequency (workload / application changes)
 - Increased time per execution (usually execution plan or data volume changes)
 - Workload profile changes (workload / application changes)
 - SQL statements that are consuming the majority of the time (See [How to Identify Resource Intensive SQL for Tuning](#) for more information).
See the section below if regression is due to a SQL statement and consider...
 - Was an application change or upgrade also performed?
 - Are there resource issues (i.e. cpu queues will inflate wait times)?
 - Has the data volume changed?
 - Were there any other changes done - filesystem to ASM, single instance to RAC,

etc.?

2. If Regression is Due to a SQL Execution Plan Change...

If you are licensed for the Tuning Pack, use the [SQL Tuning Advisor](#) to tune the SQL. Otherwise:

1. Find the execution plans from the old and the new releases (you can use `sprepsql.sql` to query the statspack schema for details of a particular SQL statement). Please see [Recording Explain Plans on 9i before an upgrade to 10g](#) for details on how to do this.
2. Investigate what changed between plans (and review the Behavior Changes tab); See the table below for typical causes and solutions.
3. Implement a solution to ensure you get the good execution plan.

Some common causes and solutions are:

Possible Cause	Possible Solutions
Suboptimal statistics collection	Review statistics collection methods, consider larger sample sizes, non-default or auto settings
Lack of system statistics	Collect system stats using the DBMS_STATS. GATHER_SYSTEM_STATS package
Rule-based Optimizer to Cost-based Optimizer Migration	See Metalink Note ID 222627.1, Migrating to the Cost-Based Optimizer

Additional Suggestions

- Use an outline to obtain the desired plan, but make sure you have applied patches for [5893396](#), [5959914](#), and [6114166](#).
- As a last resort, use a hint to change the plan (if you know what kind of plan you want).

For additional SQL tuning suggestions, please consult:

- [Oracle Performance Tuning Guide, SQL Tuning Overview](#)
- [Support's Oracle Performance Diagnostic Guide, Query Tuning](#)
- [MetaLink Knowledge Browser > Query Tuning](#)

3. *If Regression is Due to Something Else*

1. *Look for a bug in the [10.2.0.4 patch list](#) .*
2. *Consult the [Oracle Performance Tuning Guide](#) and Support's [Oracle Performance Diagnostic Guide](#) for additional help.*

When All Else Fails...Going Back to the Previous Version

If the production system is unstable or performing poorly and there is no more time to troubleshoot the issue, it might be time to consider downgrading or implementing a fallback plan to return to the previous version.

Downgrading

Downgrading brings the database back to the version prior to the upgrade. **You cannot downgrade if you have changed the COMPATIBLE parameter to 10.2.0.x from 9.2.0.x.** A discussion on downgrading is found in the *Prepare and Preserve* step and in the [Oracle Database Upgrade Guide 10g Release 2 \(10.2\)](#).

Executing a Fallback Plan

If downgrading is not possible or undesirable, then a fallback plan may be chosen to return to the pre-upgrade state. Discussion of fallback plans can be found in the *Prepare and Preserve* step. Execute the fallback plan that suits your business need and plans you've made.

Obtaining Support

Searching for Content on Metalink

Please see the Knowledge Browser for Upgrade and Migrations areas at [Getting Started](#) and [Upgrading to V10g](#).

Logging an SR After an Upgrade

Please see the following notes for suggestions on logging SRs properly:

[Note 210014.1 How to Log a Good Performance Service Request](#)

[Note.339834.1 Gathering Information for RDBMS Performance Tuning issue](#)

[Note.68735.1 Diagnostics for Query Tuning Problems](#)

[Note.166650.1 Working Effectively With Global Customer Support](#)

Behavior Changes

This section documents important changes in behavior between Oracle9i Release 2 (9.2) and Oracle Database 10g Release 2 (10.2). This section focuses on behavior changes that require a DBA to make an informed decision to minimize the risks that may be introduced by the changes. This section does not describe all changed behavior or new features in Oracle Database 10g. For a complete list of all new features introduced in Oracle Database 10g, see the [Oracle Database New Features Guide 10g Release 2 \(10.2\)](#)

This page is an accumulation of real-world knowledge and experience obtained from Support and Development engineers and working with Oracle customers on different upgrade scenarios. Pay careful attention to these Behavior Changes to avoid the most common issues when upgrading from Release 9.2 to Release 10.2.

Architecture

Optimal Flexible Architecture

Prior to Oracle Database 10g, the Optimal Flexible Architecture (OFA) standard recommended Oracle home path was similar to the following:

```
/u01/app/oracle/product/9.2.0
```

For Oracle Database 10g, the OFA recommended Oracle home path is now similar to the following:

```
/u01/app/oracle/product/10.2.0/type[_n]
```

type is the type of Oracle home (for example, Oracle Database (db) or Oracle Client (client)) and is an optional counter. This syntax provides the following benefits:

You can install different products with the same release number in the same Oracle base directory, for example:

```
/u01/app/oracle/product/10.2.0/db_1  
/u01/app/oracle/product/10.2.0/client_1
```

You can install the same product more than once in the same Oracle base directory, for example:

```
/u01/app/oracle/product/10.2.0/db_1  
/u01/app/oracle/product/10.2.0/db_2
```

Parallel query

Parallel Query (PQ) has been enhanced in 10.2 to allow more queries to be parallelized than was possible in previous releases.

In an unrestrained environment (for example, one which has default degree of parallelism at the table or index level), more queries are parallelized than in prior releases.

The combination of having PQ unrestrained and the change in the default value for `PARALLEL_MAX_SERVERS` may result in a significant increase in workload. This is because as more queries are parallelized, there may be more parallel query slaves needed to run those queries.

Note: The default value for `PARALLEL_MAX_SERVERS` has changed to be based on the number of CPUs. This is discussed in the '`PARALLEL_MAX_SERVERS`' section under the 'Initialization Parameters' heading below.

Recyclebin

The `RECYCLEBIN` parameter is used to enable or disable the Flashback Drop feature. If the parameter is set to `OFF`, then dropped tables do not go into the recycle bin. If this parameter is set to `ON`, then dropped tables go into the recycle bin and can be recovered.

The recycle bin is actually a data dictionary table containing information about dropped objects. Dropped tables and any associated objects such as indexes, constraints, nested tables, and the like are not removed and still occupy space. They continue to count against user space quotas until specifically purged from the recycle bin or the unlikely situation where they must be purged by the database because of tablespace space constraints.

This can affect applications that need to drop large numbers of tables as more time is needed to drop the tables and more space is required.

Hash Group By aggregation enabled

Oracle Database 10g Release 2 introduces a new feature called Hash Group By Aggregation, which allows a hash algorithm to process GROUP BY statements.

The GROUP BY clause still performs sort operations, but in Oracle Database 10g, the new hashing algorithm does not guarantee the order of data retrieval and may change the order for returned rows. Thus, you should not rely on GROUP BY clause to return rows in a particular order. In previous versions, the GROUP BY clause may have returned rows in particular order. If you need rows to be returned in a particular order you need to add an ORDER BY clause to your SQL statement. See Oracle MetaLink [Note: 345048.1](#).

Optimizer

Initialization Parameters

1. *Optimizer_mode*

The OPTIMIZER_MODE parameter has a new default value of ALL_ROWS in Oracle Database 10g. This means the Optimizer will no longer use Rule Based Optimizer (RBO) when a table has no statistics. In Oracle Database 10g, the Optimizer uses dynamic sampling to get statistics for these tables and uses Cost Based Optimizer (CBO). The other two possible values are FIRST_ROWS_N and FIRST_ROWS. The CHOOSE and RULE values are no longer supported.

2. *Optimizer_dynamic_sampling*

The OPTIMIZER_DYNAMIC_SAMPLING parameter has a new default value of 2 in Oracle Database 10g. This means dynamic sampling will be applied to all unanalyzed tables even if the table has indexes on it. It also means that twice the number of blocks will be used to calculate statistics than were used in Oracle9i databases.

3. *Optimizer_secure_view_merging*

Notes

- [Upgrading from 9i to 10g - Potential Query Tuning Related Issues](#)
- [Upgrading from Oracle Database 9i to 10g: What to expect from the Optimizer](#)

When a SQL statement that refers to a view is parsed the view referenced in a query is expanded into a separate query block, which represents the view definition, and therefore the result of the view. The query transformer then merges the view query block into the query block that contains the view. In Oracle Database 10g a new parameter `optimizer_secure_view_merging` has been introduced. It has a default value of `TRUE`, which means the Optimizer will ensure that only views whose integrity will not change will be merged. In other words the optimizer will check to ensure that view merging does not violate any security intentions of the original view creator. For instances, if a query uses a user-defined functions that belong to a user who is different from the owner of the objects accessed in the view, then `OPTIMIZER_SECURE_VIEW_MERGING` will prevent any kind of view merge and predicate push down for this query. With the new security checks it's possible that a view, which was merged in Oracle9i, may not be merged in Oracle Database 10g. However, if you do not have any security concerns with your application you can disable the additional checks and revert to the Oracle9i behavior by setting `OPTIMIZER_SECURE_VIEW_MERGING` to `FALSE`.

Alternatively, you can grant `MERGE ANY VIEW` privilege to specific users and leave the setting of `OPTIMIZER_SECURE_VIEW_MERGING` to `TRUE`.

Cost Based Transformations

Oracle transforms SQL statements using a variety of sophisticated techniques during query optimization. The purpose of this phase of query optimization is to transform the original SQL statement into a semantically equivalent SQL statement that can be processed more efficiently. In Oracle9i, the following Optimizer transformation were heuristic based:

- Complex View Merging
- Subquery Unnesting
- Join Predicate Push Down

This means the transformations were applied to incoming SQL statements based on the structural properties of the query: number of tables, types of joins and filters, presence of grouping clauses, etc. However, the selectivity, cardinality, join order, and other related costs of various database operations, were not taken into account.

In Oracle Database 10g Release 2, a new general framework for cost-based query transformations was introduced, so these transformations are now cost-based transformation. This means the queries are rewritten or transformed as they were before and its cost is estimated. This process is repeated multiple times applying a new set of transformations each time. The Optimizer then selects the best execution plan based on the one with the lowest cost.

Rule-Based Optimizer

The rule-based optimizer (RBO) is not supported in Oracle Database 10g. Oracle Database 10g only supports the cost-based optimizer (CBO). See Oracle MetaLink [Note: 189702.1](#) for more detail.

New automatic statistics gathering job

In Oracle Database 10g Release 2, an automatic statistics gathering job is enabled by default when a database is created, or when a database is upgraded from an earlier database release. Oracle database will automatically collect statistics for all database objects which are missing statistics by running an Oracle Scheduler job (GATHER_STATS_JOB) during a predefined maintenance window. The maintenance window opens every night from 10:00 pm to 6:00 am and all day on weekends.

The job gathers optimizer statistics by calling the DBMS_STATS.

GATHER_DATABASE_STATS_JOB_PROC procedure. This is an internal procedure, but it operates in a very similar fashion to the DBMS_STATS.

GATHER_DATABASE_STATS procedure using the GATHER AUTO option. The primary difference is that the DBMS_STATS.

GATHER_DATABASE_STATS_JOB_PROC procedure prioritizes the database objects that require statistics, so that those objects, which most need updated statistics, are processed first. You can verify that the automatic statistics gathering job exists by viewing the DBA_SCHEDULER_JOBS view.

```
SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME =
'GATHER_STATS_JOB';
```

Statistics on a table are considered stale when more than 10% of the rows are changed (total # of inserts, deletes, updates) in the table. Oracle monitors the DML activity for objects and record it in the SGA. The monitoring information is periodically flushed to disk and is exposed in *_tab_modifications view.

```
SELECT TABLE_NAME, INSERTS, UPDATES, DELETES FROM
USER_TAB_MODIFICATIONS;
```

The automatic statistics gathering job uses the default parameter values for DBMS_STATS procedures. If you wish to change these default values you can use the DBMS_STATS.SET_PARAM procedure. To change the 'ESTIMATE_PERCENT' you can use:

```
BEGIN
DBMS_STATS.SET_PARAM('ESTIMATE_PERCENT','5');
END;
/
```

If you already have a well established statistics gathering procedure or if for some other reason you need to disable automatic statistics gathering altogether, the most direct approach is to disable the GATHER_STATS_JOB as follows:


```

BEGIN
DBMS_SCHEDULER.DISABLE('GATHER_STATS_JOB');
END;
/

```

There was no default automatic statistics gathering job in Oracle9i.

Changes in default values for DBMS_STATS parameters

Changes in the default values for DBMS_STATS parameters

The default value for a number of the parameters used in the DBMS_STATS procedure gather statistics subprograms have change in Oracle Database 10g. Table 1 below highlights these changes.

Parameter	9.2 Value	10.2 Value
METHOD_OPT	FOR ALL COLUMNS SIZE 1	FOR ALL COLUMNS SIZE AUTO
ESTIMATE_PERCENT	100 (Compute)	DBMS_STATS. AUTO_SAMPLE_SIZE
GRANULARITY	DEFAULT (Table & Partition)	AUTO
CASCADE	FALSE	DBMS_STATS. AUTO_CASCADE
NO_INVALIDATE	FALSE	DBMS_STATS. AUTO_INVALIDATE

Table 1 Default values for parameters used in DBMS_STATS

The **METHOD_OPT** parameter controls the creation of histograms during statistics creation. With the new default value of "FOR ALL COLUMNS SIZE AUTO", Oracle automatically determines which columns require histograms and the number of buckets that will be used. A column is a candidate for a histogram if it has been seen in a predicate (equality, range, LIKE, etc). Oracle will verify whether the column is skewed before creating a histogram.

The **ESTIMATE_PERCENT** parameter determines the percentage of rows used to calculate the statistics. In Oracle9i the default percentage was 100% or all of the rows in the table. However, in Oracle Database 10g, statistics are gathered using the sampling method. Oracle automatically determines the appropriate sample size for every table in order to get accurate statistics.

The **GRANULARITY** parameter dictates at which level statistics will be gathered. The possible levels are table (global), partition, or subpartition. With the new default setting of "AUTO", Oracle database will determine the granularity based on the objects partitioning

type.

The **CASCADE** parameter determines whether or not statistics are gathered for the indexes on a table. In Oracle Database 10g it is set to "DBMS_STATS.AUTO_CASCADE" by default, which means Oracle will determine whether or not index statistics need to be collected.

In Oracle9i the **NO_INVALIDATE** parameter determined if the dependant cursors would be invalidated immediately after the statistics were gather or not. With the new setting of "DBMS_STATS.AUTO_INVALIDATE", Oracle decides when to invalidate dependent cursors.

You can see the default values using the following queries:

```
select dbms_stats.get_param('method_opt') from dual;
select dbms_stats.get_param('estimate_percent') from dual;
select dbms_stats.get_param('granularity') from dual;
select dbms_stats.get_param('cascade') from dual;
```

```
SQL> select dbms_stats.get_param('ESTIMATE_PERCENT') from dual;
```

```
DBMS_STATS.GET_PARAM('ESTIMATE_PERCENT')
```

```
-----
```

```
DBMS_STATS.AUTO_SAMPLE_SIZE
```

To change any of the default values, use the DBMS_STATS.SET_PARAM procedure.

```
exec dbms_stats.set_param('METHOD_OPT', 'FOR ALL COLUMNS SIZE 1');
```

```
SQL> select dbms_stats.get_param('method_opt') from dual;
```

```
DBMS_STATS.GET_PARAM('METHOD_OPT')
```

```
-----
```

```
FOR ALL COLUMNS SIZE 1
```

System Statistics

In Oracle9i system statistics were introduced to enable the CBO to effectively cost each operation in an execution plan by using information about the actual system hardware executing the statement, such as CPU speed and IO performance. However, if system statistics were not gathered in Oracle9i the CBO would revert back to the Oracle8i costing model. In Oracle Database 10g Release 2 the use of system statistics is on by default and system statistics are automatically initialized to:

```
ioseektim = 10ms
iotrfspeed = 4096 bytes/ms
cpuspeednw = gathered value, varies based on system
```

When you gather system statistics in Oracle Database 10g they will override these initial values. To gather system statistics you can use DBMS_STATS.GATHER_SYSTEM_STAT.

At the beginning of the peak workload window execute the following command:

```
BEGIN
DBMS_STATS.GATHER_SYSTEM_STATS( ' START ' ) ;
END ;
/
```

At the end of the peak workload window execute the following command:

```
BEGIN
DBMS_STATS.GATHER_SYSTEM_STATS( ' END ' ) ;
END ;
/
```

Statistics on Dictionary Tables

Since the default value for optimizer_mode in Oracle Database 10g forces the use of the CBO, all tables in the database need to have statistics, including all of the dictionary tables. During the upgrade process Oracle will automatically gather statistics on the dictionary tables. These statistics will be maintained via the automatic statistics gathering job run during the maintenance window. If you choose to switch off the automatic statistics gathering job for user schema, consider leaving it on for the dictionary tables. You can do this by changing the value of AUTOSTATS_TARGET to ORACLE instead of ALL using DBMS_STATS.SET_PARAM.

```
BEGIN
DBMS_STATS.SET_PARAM(AUTOSTATS_TARGET, ' ORACLE ' ) ;
END ;
/
```

Statistics on Fixed Objects

You will also need to gather statistics on dynamic performance tables (fixed objects) such as V\$SQL due to the new default value for optimizer_mode. It is important to gather statistics on the fixed objects as they are often queried to supply information to STATSPACK and the new Automatic Workload Repository in Oracle Database 10g and you need to give the CBO accurate statistics for these objects. You only need to gather statistics once for each workload. You can collect statistics on fixed objects using DBMS_STATS.GATHER_FIXED_OBJECT_STATS.

```
BEGIN
DBMS_STATS.GATHER_FIXED_OBJECT_STATS;
END;
/
```

Statistics on Indexes

In Oracle Database 10g Release 2 statistics are automatically gathered for an index at the time of creation. You no longer have to specify the compute statistic clause as part of the create index command as you did in earlier releases.

Restoring Statistics

When you gather statistics they are automatically published as soon as the gathering process has completed. However, it may become necessary to revert back to the statistics you had before the gathering process. In Oracle Database 10g when you gather statistics, the original statistics are automatically kept as a backup and can be easily restored by running DBMS_STATS.RESTORE_TABLE_STATS.

```
RESTORE_TABLE_STATS.
```

```
BEGIN
DBMS_STATS.RESTORE_TABLE_STATS ( ' CASCADE ' , ' DBMS_STATS .
AUTO_CASCADE ' ) ;
END;
/
```

DBMS_STATS AND PARALLEL DEGREE

In DBMS_STATS subprograms the parameter DEGREE controls the parallel degree for the operation. In Oracle Database 10g Release 2 there is a new threshold value which Parallelism will start for DBMS_STATS subprograms. The minimum number of slaves is 3 and the minimum blocks per PX slave should be at least 1000. So, the table must have 3000 blocks before DBMS_STATS subprograms will consider executing in parallel, otherwise the DEGREE parameter will be ignored. In Oracle9i Release 2 or earlier there was no such restriction. This will affect only tables with less than 3000 blocks.

Bind peeking

With bind peeking the Optimizer peeks at the values of user-defined bind variables on the first execution of a query (during hard parse). The Optimizer determines the execution plan based on the initial value of bind variables. On subsequent invocations of the query, no peeking takes place, so the original execution plan is used by all future executions, even if the value of the bind variables change. The presence of a histogram on the column used in the expression with the bind variable may cause a different execution plan to be generated for the statement depending the initial value of the bind variable being peeked. This could cause unpredictable query performance because the execution plan varies depending on the values of the bind variables on its first invocation. It is possible you may run into this issue in Oracle Database 10g due to the change in the default behavior in DBMS_STATS (see the information on the method_opt parameter in the "Changes in the default values for DBMS_STATS parameters" section) even though you never experienced it in Oracle9i. If this is the case then you can regather statistics on the table without histograms or change the default value of method_opt parameter. As a last resort you can disable bind peeking by setting the parameter _optim_peek_user_binds to false.

Initialization Parameters

SHARED_POOL_SIZE

In previous releases, the amount of shared pool memory that was allocated was equal to the value of the SHARED_POOL_SIZE initialization parameter plus the amount of internal SGA overhead computed during instance startup. Starting with Oracle Database 10g, the value of SHARED_POOL_SIZE must now also accommodate this shared pool overhead.

Please refer to [Oracle Database Upgrade Guide 10g Release 2 \(10.2\)](#)

SHARED_POOL_SIZE and Automatic Storage Management(ASM): On a database instance using ASM, additional memory is required to store extent maps.

Please refer to [Minimum for SHARED_POOL_SIZE Parameter in 10.2 Version](#)

SESSION_CACHED_CURSORS

In previous Oracle Database releases, the number of SQL cursors cached by PL/SQL was determined by the OPEN_CURSORS initialization parameter. Starting with Oracle Database 10g, the number of cached cursors is determined by the SESSION_CACHED_CURSORS initialization parameter.

LOG_ARCHIVE_FORMAT

Starting with Oracle Database 10g, if the COMPATIBLE initialization parameter is set to 10.0.0 or higher, then archive log file names must contain each of the elements %s (sequence), %t (thread), and %r (resetlogs ID) to ensure that all archive log file names are unique. If the LOG_ARCHIVE_FORMAT initialization parameter is set in the parameter file, then make sure the parameter value contains the %s, %t, and %r elements.

PGA_AGGREGATE_TARGET

Starting with Oracle Database 10g, Automatic PGA Memory Management is now enabled by default (unless PGA_AGGREGATE_TARGET is explicitly set to 0 or WORKAREA_SIZE_POLICY is explicitly set to MANUAL). PGA_AGGREGATE_TARGET defaults to 20% of the size of the SGA, unless explicitly set. Oracle recommends tuning the value of PGA_AGGREGATE_TARGET after upgrading.

Until Oracle9i Release 2, PGA_AGGREGATE_TARGET parameter controls the sizing of workareas for all dedicated server connections, but it has no effect on shared server (aka MTS) connections and the *_AREA_SIZE parameters will take precedence in this case. In Oracle Database 10g, PGA_AGGREGATE_TARGET controls workareas allocated by both dedicated and shared connections.

QUERY_REWRITE_ENABLED

The default value of the initialization parameter QUERY_REWRITE_ENABLED has changed. By default it is TRUE in Oracle Database 10g and above. Prior to Oracle Database 10g the default is FALSE.

REMOTE_LOGIN_PASSWORDFILE

There are multiple modes to which a remote_login_passwordfile can be set. The different modes are SHARED, EXCLUSIVE and NONE. A SHARED password file can be used by multiple databases running on the same server, or multiple instances of a Real Application Clusters (RAC) database. A SHARED password file cannot be modified which means that one cannot add users to a SHARED password file. Any attempt to do so or to change the password of SYS or other users with the SYSDBA or SYSOPER privileges generates an error.

In Oracle9i Release 2 the default value of REMOTE_LOGIN_PASSWORDFILE is EXCLUSIVE. In 10.2, the default value of this parameter is set to SHARED.

PARALLEL_ADAPTIVE_MULTI_USER

The adaptive multiuser feature adjusts the degree of parallel(DOP) for an operation based on user load. For example, you might have a table with a DOP of 5. This DOP might be acceptable with 10 users. However, if 10 more users enter the system and `PARALLEL_ADAPTIVE_MULTI_USER` is set to true, Oracle automatically reduces the DOP to spread resources more evenly according to the perceived Oracle load.

In Oracle Database 10g the default value of `parallel_adaptive_multi_user` is true so Oracle will always automatically reduce the DOP of an operation as the load increases on the system. In Oracle9i the default value of `parallel_adaptive_multi_user` was derived from `parallel_automatic_tuning` which defaults to false, so `parallel_adaptive_multi_users` was also false. If `parallel_automatic_tuning` was true in 9i Oracle would set the value of `parallel_adaptive_multi_user` parameter to true. Note `parallel_automatic_tuning` has been deprecated in Oracle Database 10g.

PARALLEL_MAX_SERVERS

The Default value of `PARALLEL_MAX_SERVERS` has changed in Oracle Database 10g

In Oracle Database 10g, the default is derived using the following formula:

$$\text{CPU_COUNT} \times \text{PARALLEL_THREADS_PER_CPU} \times (2 \text{ if } \text{PGA_AGGREGATE_TARGET} > 0; \text{ otherwise } 1) \times 5$$

In Oracle9i Release 2

If `PARALLEL_AUTOMATIC_TUNING` is false, the default value of `PARALLEL_MAX_SERVERS` is 5.

If `PARALLEL_AUTOMATIC_TUNING` is TRUE, the default value of `PARALLEL_MAX_SERVERS` is CPU x 10.

SKIP_UNUSABLE_INDEXES

The default value of `SKIP_UNUSABLE_INDEXES` is `TRUE` in Oracle Database 10g. `SKIP_UNUSABLE_INDEXES` enables or disables the use and reporting of tables with unusable indexes or index partitions. In earlier releases prior to Oracle Database 10g, `SKIP_UNUSABLE_INDEXES` was a session parameter only. In Oracle Database 10g and later, it is now an initialization parameter and defaults to `true`. The `true` setting disables error reporting of indexes and index partitions marked `UNUSABLE`. This setting allows all operations (inserts, deletes, updates, and selects) on tables with unusable indexes or index partitions.

Performance and Monitoring

Statistics_level

The default `STATISTICS_LEVEL` is `TYPICAL` in both Oracle9i and Oracle Database 10g. In Oracle Database 10g, this parameter value also enables the following:

- Global monitoring for segments which do not have statistics, or whose statistics are stale.
- New manageability features, including the Automatic Workload Repository (AWR), Automatic Database Diagnostic Monitor (ADDM).

Automatic Shared Memory Management (ASSM) also requires `STATISTICS_LEVEL` to be `TRUE` for this feature to be enabled.

Explain Plan Enhancements

In Oracle Database 9i the PL/SQL package `DBMS_XPLAN` was introduced to provide an easier way to format the output of the `EXPLAIN PLAN` command. In Oracle Database 10g the `DBMS_XPLAN` package has been extended to enable you to display execution plans from three additional sources:

- `V$SQL_PLAN`
- Automatic Workload Repository (AWR)
- SQL Tuning Set (STS)

Each of the `DBMS_XPLAN.DISPLAY` functions takes a `format` parameter. The valid parameter values are `basic`, `typical`, `all`. The `format` parameter controls the amount of detail in the plan output, from a high level summary that only includes the execution plan (`format=>'basic'`), to finer grained detail (`format=>'all'`). The default is `'typical'`. In Oracle 10g, additional options can also be passed with the `format` parameter to selectively display the detailed information, such as

predicates used, the value of the bind variables used to generate the execution plan. DBMS_XPLAN can also be used to display extended plan statistics if the data is available.

Materialized View Refresh

In 10.2 Complete refresh of single materialized view using `dbms_mview.refresh` performs a delete of the materialized view base table instead of a truncate (prior to 10.2 Materialized view used truncate). This may require more Time to perform the complete refresh and generate more redo.

This change was made to prevent wrong results. Refreshing a single MV is not atomic even if `ATOMIC_REFRESH = true` (the default). This can lead to wrong results in sessions querying the materialized view, as the row count can suddenly change to 0 (as the refresh truncates the MV). An atomic refresh should not affect read consistency in this way.

Table Lock

'SELECT FOR UPDATE' operations now take a TM lock in 'Row Exclusive' mode (i.e. SX or mode=3 in V\$LOCK).

For applications that have unindexed foreign keys, this means UPDATE or DELETE operations on a PARENT table row will now be BLOCKED by any active 'SELECT FOR UPDATE' on the child table, even if the child row is for a different parent key to that being deleted or updated.

If this change is undesirable on your site (for example, if you have many unindexed foreign keys), please refer to [bug 4969880](#) for how to revert back to the original behavior.

Administration

CONNECT Role Privileges

In Oracle Database 10g Release 2 (10.2) the CONNECT role only contains the CREATE SESSION privilege. Making this change enables new and existing database customers to enforce good security practices more easily. Applications that rely on the CONNECT role to create tables, views, sequences, synonyms, clusters, or database links, or applications that use the ALTER SESSION command dynamically, may fail due to insufficient privileges.

FAILED_LOGIN_ATTEMPTS

As of Oracle Database 10g Release 2, the limit for FAILED_LOGIN_ATTEMPTS for the DEFAULT profile is 10.

Prior to Release 10.2, the default was UNLIMITED.

Streams

Initialization Parameters

1. AQ_TM_PROCESSES

This parameter rules the behavior of Time Monitor background process. The behavior of this parameter has radically changed between releases 9iR2 and 10GR2.

In Oracle9i Release 2, it was recommended to set this parameter to a value between 1 and 10. The number of Time Monitor processes created was equal to the setting of AQ_TM_PROCESSES. Time Monitor will not start if the value is set to 0.

In Oracle Database 10g Release 2, it is highly recommended to unset this parameter. This will allow the Time Monitor processes to work in autotune mode. The Time Monitor coordinator process will get created on database startup and it will spawn time monitor slaves processes as needed. To unset this parameter, it should be removed from the pfile or spfile. Please see MetaLink [Note 428441.1](#) (Warning: "Aq_tm_processes Is Set To 0" Message in Alert Log After Upgrade to 10.2.0.3) for more information. Setting the value of AQ_TM_PROCESSES to 0 will disable autotune mode.

Since the introduction of buffered messages on Oracle Database 10g Release 2, it is highly recommended not to set AQ_TM_PROCESSES to 10, as this will disable the AQ time based operations on buffered messages leading to a situation where memory gets exhausted. Applications such as Oracle Streams, Oracle Alerts, and Oracle Datapump make use of the buffered messages feature.

Architectural Changes

Several architectural changes have been introduced in Oracle Database 10g Release 2 compared to release Oracle9i Release 2.

1. Memory Management

In Oracle9i Release 2 the memory used by Streams processes was directly acquired from the shared pool memory and by default Streams processes were able to consume up to 10% of the shared pool memory. It was possible to change the percentage of memory consumed in the shared pool by using the hidden parameter `_first_spare_parameter`.

In Oracle Database 10g Release 2 the memory used by Streams processes is reserved through the new parameter called `streams_pool_size`. The minimum recommended value for `streams_pool_size` is 200M. Streams pool is auto-tunable if the the following are set:

- ***`SGA_TARGET > 0`***
- ***`streams_pool_size=0`***
- ***`shared_pool_size=0`***

Due to this change hidden parameters `_kgghdsidx_count` and `_first_spare_parameter` are not required anymore.

2. Automatic Flow Control

In Oracle9i Release 2 it was recommended to create a job that through a procedure frequently monitors the consumption of messages in memory and stops the capture process to avoid excessive spill over. This mechanism was called flow control.

In Oracle Database 10g Release 2, the flow control process is now internalized and automatic and does not required any setup. It will pause the capture to avoid excessive spill over.

3. Logminer Checkpointing

In Oracle9i Release 2, it was recommended to create a job that through a procedure that frequently purges old logminer checkpoint metadata.

In Oracle Database 10g Release 2, purge logminer checkpoint metadata is an automatic process controlled by the capture parameter `checkpoint_retention_time`. Also it is recommended to increase the frequency of the checkpoints to 100Mb through the capture parameter `_checkpoint_frequency`.

4. Supplemental Logging

In Oracle9i Release 2, supplemental logging must be configured for replicated tables with Primary Keys, Unique Indexes, and Foreign Keys. This supplemental logging must be explicitly configured on the source database in 9.2.

In Oracle Database 10g Release 2, additional key words have been provided to simplify table level supplemental logging configuration. It is now possible to specify PRIMARY KEY, FOREIGN KEY, UNIQUE, or ALL columns without managing the columns in a supplemental log group. Supplemental logging is generated automatically in Oracle Database 10g Release 2 for Primary Keys, Unique Indexes, and Foreign Key columns.

However, Oracle9i Release 2 apply processes require additional logging for tables with Unique Indexes, and Foreign Keys if apply parallelism is employed at the 9.2 database. This supplemental logging must be explicitly configured on the source database in 10.2.

5. Name of processes

In Oracle9i Release 2:

- ***Capture processes were called CPXX.***
- ***Apply processes were called APXX.***

In Oracle Database 10g Release 2:

- ***Capture processes are called CXXX and may use as many Parallel slave processes as required.***
- ***Apply processes are called AXXX and may use as many Parallel slave processes as required.***

Patches Recommended

This page will help you prepare for an upgrade by providing a list of patches that are recommended for your platform. This list of patches is based on the most common issues encountered by customers on version 10.2.0.3 of the database. To locate recommended patches required for your Oracle Database 10g upgrade, expand the list of recommended patches associated with the operating system in which the upgrade is being performed. **Carefully review each patch and only apply the patches specific to your environment and the features being used.** Each table contains the following:

- **Bug Details:** The bug number for the patch. Click the link to view the bug details. If a bug is not published externally, a link will not be provided.
- **Subject:** The abstract from the bug. Hover the mouse over the subject to see a description of the bug if it is available.
- **Type:** The type of patch (Merge patch or Standalone patch).
- **Component:** The database component to which the patch applies.
- **Patch:** The patch number. Click the link to view the patch download page.

Operating System

This section recommends patches for each of the following specific operation systems.

AIX 5L Based Systems (64-Bit)

AIX 5L Based Systems (64-Bit)				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	9503316
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	9403059
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	9290833
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	Available Soon
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9440106
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9376243
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9054497
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9023535
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9188713

5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS. GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9132104
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9266039
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	9309373
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	9440772
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9420475
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9596919
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9823011
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	Available Soon
Patch	Subject	Type	Component	Patch

HP-UX Itanium

HP-UX Itanium				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	9369391
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	9232075
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	9444234
5577046	ADD or DROP attribute causes UNION query to fail with ORA- 1790	Standalone patch	RDBMS	10009248
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9235189
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9364796
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9392724
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9197203
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9223810
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS. GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9226034
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9754481
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	9357559

6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10050661
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9409837
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9613152
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9901071
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10050652
Patch	Subject	Type	Component	Patch

HP-UX PA-RISC

HP-UX PA-RISC (64-Bit)				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	9217492
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	9179634
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	9292968
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	10022319
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9436607
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9051485
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9436615
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9056728
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9145279
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9163048
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9515817
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	10021107
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10050670
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9409964
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9588213
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9840941

6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10050756
Patch	Subject	Type	Component	Patch

Linux Itanium

Linux Itanium				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	10010305
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	10010019
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	10009464
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	10010876
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9225934
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9223601
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	10050811
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9280188
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9491424
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9223528
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	10050793
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	10021112
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10050794
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9410180
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	10013551
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	10013670
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10052797
Patch	Subject	Type	Component	Patch

Linux x86

Linux x86				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	10006212
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	10006279
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	10006213
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	10006211
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9516683
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	8936772
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9079411
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9513903
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9354002
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9102364
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9179874
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	9061297
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10008057
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9403421
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9596992
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9894838
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10005987
Patch	Subject	Type	Component	Patch

Linux x86-64

Linux x86-64 (AMD64/EM64T)				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	9466981
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	9005083
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	9326505
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	10050772
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9434605
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9026389
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9105373
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9047409
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9185815
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9162583
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9594036
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	9358879
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10050771
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9410662
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9640527
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9832364
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10050651
Patch	Subject	Type	Component	Patch

Solaris Operating System (SPARC 64-Bit)

Solaris Operating System (SPARC 64)				
Bug Details	Subject	Type	Component	Patch
4899479	UNDO/REDO CORRUPTION WITH STORED PROCEDURE	Standalone patch	RDBMS	9401274
5363584	ARRAY INSERT INTO PARTITIONED TABLE CAN CORRUPT REDO	Standalone patch	RDBMS	9041495
5514109	ORA-600 [KQL-HASH-COLLISION] AND ORA-955 AFTER APPLYING CPU	Standalone patch	RDBMS	9358735
5577046	ADD or DROP attribute causes UNION query to fail with ORA-1790	Standalone patch	RDBMS	9728311
5605370	APPLIED BDL 6 AND 4966417 - HITTING ORA-600S [KOKCUP:01] [KSSADPM:NULL PARENT]	Standalone patch	RDBMS	9042687
5728380	STMAIL: 2 PROCESSES SPIN ON INSERT STATEMENT	Standalone patch	RDBMS	9052089
5765958	ORA-600 [QCSFBDNP:1]/[QCSCPQBTEXT]	Standalone patch	RDBMS	9038772
5892355	ASSERT IN KOPEPIC THROWING 22635 INTERNAL ERROR NEEDS TO BE REMOVED	Standalone patch	RDBMS	9026031
5896963	PERF REGRESS LGWR KSLPSTEVENT	Standalone patch	RDBMS	9145030
5907779	"CURSOR: PIN S WAIT ON X" RUNNING DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC	Standalone patch	RDBMS	9102810
5933477	CLIENT APPLICATION ENDED WITH CORE DUMP AFTER FLUSH SHARED_POOL.	Standalone patch	RDBMS	9186188
5949701	MERGE LABEL REQUEST ON TOP OF 10.2.0.3 FOR BUGS 5648872 5863277	Merge patch	RDBMS	9137893
6017420	AROLTP-C:FG ENCOUNTERED ORA-00600 [KCBO_LINK_Q_1]	Standalone patch	RDBMS	10008190
6128197	ORA-00600: INTERNAL ERROR CODE, ARGUMENTS: [KCRFR_RESIZE2], [67108864]	Standalone patch	RDBMS	9423284
6455161	HIGHER "CONSISTENT GETS" AFTER TRUNCATE	Standalone patch	RDBMS	9630223
6646613	IOT CORRUPTION / ORA-600 [KCBGTCR_1] AFTER UPGRADE FROM 9.2 -> 10G	Standalone patch	RDBMS	9739759
6869828	MERGE LABEL REQUEST FOR THE FIXES FOR BUGS 5903829, 6122097 ON TOP OF 10.2.0.3	Merge patch	RDBMS	10006453
Patch	Subject	Type	Component	Patch

Microsoft Windows

Oracle provides mini patches for Microsoft Windows. New mini patches are provided on a regular basis. To download the latest mini patch perform the following steps:

1. Login to MetaLink
2. Click on the "Patches & Updates" tab
3. Click on "Advanced Search"
4. Set each field to the following:

Field	Value
Product or Product Family	RDBMS Server (Oracle Database Family)
Release	Oracle 10.2.0.3
Compatible With	leave as is
Platform or Language	Microsoft Windows (32-bit) or other Windows platform that is applicable to your server.
Patch Type	Any
Priority	Any

5. Click "Go"

A list will be provided. The newer mini patches should display toward the top of the the list. Look for a patch name similar to: ORACLE 10.2.0.3 PATCH 16 BUG FOR MICROSOFT WINDOWS 32BIT. In this example the mini patch is #16. The number will increase each time a new mini patch is released.

NOTE: It is recommended to apply the most current mini patch.

Documentation

Documentation

[Oracle Database Upgrade Guide 10g Release 2 \(10.2\)](#)

This guide is intended for database administrators (DBAs), application developers, security administrators, system operators, and anyone who plans or executes Oracle Database upgrades. Oracle Database Upgrade Guide contains information that describes the features and functionality of the Oracle Database (also known as the standard edition) and the Oracle Database Enterprise Edition products. The Oracle Database and the Oracle Database Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are optional.

[Oracle Database Performance Tuning Guide](#)

This guide is an aid for people responsible for the operation, maintenance, and performance of Oracle. This book describes detailed ways to enhance Oracle performance by writing and tuning SQL properly, using performance tools, and optimizing instance performance. It also explains how to create an initial database for good performance and includes performance-related reference information. This book is useful for database administrators, application designers, and programmers.

[Oracle Database Readme - 10g Release 2](#)

This Readme file is relevant only to the delivered Oracle Database 10g Release 2 (10.2) product and its integral parts, such as SQL, PL/SQL, the Oracle Call Interface (OCI), SQL*Loader, Import/Export utilities, and so on. This Readme documents differences between the server and its integral parts and its documented functionality, as well as known problems and workarounds. Operating system releases, such as UNIX and Windows, often provide readme documents specific to that operating system. Additional Readme files may also exist. This Readme file is provided in lieu of system bulletins or similar publications.

[Oracle Administrator Reference 10g Release 2 for UNIX Based Operating Systems](#)

This guide is intended for anyone responsible for administering and configuring Oracle Database 10g Release 2 (10.2). If you are configuring Oracle Real Application Clusters (Oracle RAC), then refer to Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide.

[Oracle Database Installation 10g Release 2 \(10.2\) List](#)

These guides describe how to install and configure Oracle Database 10g Release 2 (10.2). Specifically, review Chapter 1 for Upgrade Recommendations. These guides are intended for anyone responsible for installing Oracle Database 10g Release 2 (10.2).

Related Documentation

[Oracle 10g Release 2 \(10.2\) Document Library](#)

This library is a list of all the Oracle Database Documentation. Here you can research new information, look up reference information, and search across the entire library.

[Oracle Database New Features Guide](#)

This guide is addressed to people familiar with previous versions of Oracle Database who would like to become familiar with features, options, and enhancements that are new in this release of the database.

[Oracle Database Concepts](#)

This manual describes all features of the Oracle database server, an object-relational database management system. It describes how the Oracle database server functions, and it lays a conceptual foundation for much of the practical information contained in other manuals. Information in this manual applies to the Oracle database server running on all operating systems.

[Oracle Database Administrator's Guide](#)

This guide describes how to create and administer an Oracle Database.

[Oracle Database SQL Reference](#)

This reference contains a complete description of the Structured Query Language (SQL) used to manage information in an Oracle Database. Oracle SQL is a superset of the American National Standards Institute (ANSI) and the International Standards Organization (ISO) SQL:1999 standard.

[Oracle Database Utilities](#)

This document describes how to use the Oracle Database utilities for data transfer, data maintenance, and database administration.

[Oracle Call Interface Programmer's Guide - Introduction and Upgrading](#)

This guide contains information on upgrading to a new release of OCI.

[Oracle Data Guard Concepts and Administration - Using SQL Apply to Upgrade the Oracle Database](#)

Starting with Oracle Database 10g Release 1 (10.1.0.3), you can use a logical standby database to perform a rolling upgrade of Oracle Database 10g software. During a rolling upgrade, you can run different releases of an Oracle database on the primary and logical standby databases while you upgrade them, one at a time, incurring minimal downtime on the primary database.

[Oracle Universal Installer and OPatch User's Guide - 10g Release 2 \(10.2\) for Windows and UNIX](#)

This chapter is designed to aid the developers, administrators, and all other users who install Oracle software (Oracle Database 10g Release 2 (10.2) for Windows and UNIX) in understanding the key concepts involved in Oracle Universal Installer.

Database Features Documentation

[Oracle Streams Concepts and Administration - Online Database Upgrade with Streams](#)

This describes how to perform a database upgrade of an Oracle Database with Oracle Streams.

[Oracle Database Net Services Reference - Upgrade Considerations for Oracle Net Services](#)

This describes coexistence and upgrade issues for Oracle Net Services.

[Oracle Data Guard Concepts and Administration - Upgrading Databases in a Data Guard Configuration](#)

The procedures in this appendix describe how to upgrade to Oracle Database 10g Release 2 (10.2) when a physical or logical standby database is present in the configuration.

[Oracle Data Guard Broker - Data Guard Broker Upgrading and Downgrading](#)

This appendix guides you through the process of upgrading or downgrading the Oracle databases and Oracle Enterprise Manager in a broker configuration.

[Oracle Spatial User's Guide and Reference - Installation, Compatibility, and Upgrade](#)

If you are upgrading to Oracle Database 10g, Oracle Spatial is automatically upgraded as part of the operation but this appendix should be reviewed for post-upgrade steps and downgrade as part of the contingency plan.

[Oracle Enterprise Manager Grid Control Installation and Basic Configuration - Enterprise Manager Upgrade](#)

This part provides the requirements and step-by-step instructions on the Enterprise Manager upgrade process.

[Oracle Database Globalization Support Guide - Upgrading the Time Zone File](#)

The time zone files that are supplied with Oracle Database 10g have been updated from version 1 to version 2 to reflect changes in transition rules for some time zone regions.